

# **PENGEMBANGAN SISTEM PENGELOLAAN REKAM MEDIS PASIEN BERBASIS WEB**

Studi Kasus: Klinik Dr. Syarif

## **SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:

Helman Taufiqurrahman

NIM: 145150200111027



PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2019

## PENGESAHAN

PENGEMBANGAN SISTEM PENGELOLAAN REKAM MEDIS PASIEN BERBASIS WEB  
STUDI KASUS: KLINIK DR. SYARIF

SKRIPSI

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :  
Helman Taufiqurrahman  
NIM: 145150200111027

Skripsi ini telah diuji dan dinyatakan lulus pada  
24 Juli 2019

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing



Tri Astoto Kurniawan, S.T., M.T., Ph.D.  
NIP: 19710518 200312 1 001

Mengetahui  
Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T., M.T., Ph.D.  
NIP: 19710518 200312 1 001

## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata di dalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 24 Juli 2019



Helman Taufiqurrahman

NIM: 145150200111027

## **PRAKATA**

Puji dan syukur penulis panjatkan ke hadirat Allah SWT karena atas berkat rahmat dan hidayah-Nya penulis dapat menyelesaikan skripsi yang berjudul “PENGEMBANGAN SISTEM PENGELOLAAN REKAM MEDIS PASIEN BERBASIS WEB. STUDI KASUS: KLINIK DR. SYARIF”. Melalui ini, penulis hendak mengucapkan terimakasih karena dalam penyusunan skripsi ini, penulis telah mendapatkan bantuan dari berbagai pihak. Untuk itu pada kesempatan ini penulis hendak mengucapkan terima kasih kepada:

1. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D. selaku Dosen Pembimbing skripsi, yang telah membina, mengarahkan, dan membimbing saya sehingga dapat menyelesaikan skripsi ini.
2. Kedua orang tua atas segala doa, semangat, dan dukungan yang selalu diberikan dengan tulus kepada penulis.
3. Teman-teman teknik informatika atas motivasi, bantuan, dan kebersamaan selama ini sehingga penulis dapat menyelesaikan skripsi dengan baik.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 24 Juli 2019

Penulis

Email: x4helman.smansa@gmail.com

## ABSTRAK

**Helman Taufiqurrahman, Pengembangan Sistem Pengelolaan Rekam Medis Pasien Berbasis Web. Studi Kasus: Klinik Dr. Syarif**

**Pembimbing: Tri Astoto Kurniawan, S.T., M.T., Ph.D.**

Klinik Dr. Syarif merupakan salah satu klinik yang berada di Kota Samarinda, Kalimantan Timur. Pengelolaan rekam medis pasien pada klinik tersebut masih dilakukan dengan menulis di kertas rekam medis dan disimpan di almari. Seiring waktu, hal tersebut dapat menyebabkan bertambahnya kebutuhan tempat untuk menyimpan rekam medis pasien. Duplikasi rekam medis dapat terjadi ketika petugas administrasi meletakkan rekam medis pada bagian almari yang salah, menyebabkan rekam medis tidak ditemukan dan petugas administrasi membuat rekam medis yang baru. Untuk mengatasi masalah tersebut, sistem pengelolaan rekam medis pasien berbasis web dikembangkan dengan tujuan untuk mempermudah proses pengelolaan rekam medis pasien. Sistem dikembangkan dengan menggunakan metode *Object Oriented Analysis & Design* (OOAD) dan metode *Object Oriented Programming* (OOP). Bahasa pemrograman yang digunakan untuk mengembangkan sistem tersebut adalah *Personal Home Page* (PHP) dan *framework* yang digunakan adalah *Codeigniter*. Pengujian unit dan pengujian integrasi dilakukan dengan menggunakan metode pengujian *white-box*, sedangkan pengujian validasi dilakukan dengan menggunakan metode pengujian *black-box*. Hasil dari pengujian unit dan pengujian integrasi menunjukkan bahwa sistem berjalan sesuai dengan rancangan yang telah dibuat. Hasil dari pengujian validasi menunjukkan bahwa seluruh kebutuhan fungsional telah terpenuhi oleh sistem. Dengan kata lain, hasil dari seluruh pengujian adalah 100% valid dan sistem layak untuk digunakan.

**Kata kunci:** Sistem informasi, pengelolaan data, rekam medis, web, berorientasi objek

## ABSTRACT

**Helman Taufiqurrahman, Development of Web-Based Management System of Patient's Medical Records. Case Study: Dr. Syarif's Clinic**

**Supervisor: Tri Astoto Kurniawan, S.T., M.T., Ph.D.**

*Dr. Syarif's Clinic is a clinic that is located in Samarinda City, East Borneo. The management of the patient's medical records in that clinic is still done by writing on a paper and stored inside the cupboard. As time goes on, it increases the place's needs for saving the medical records. Duplication of medical records could happen when the administration staff puts the medical records at the wrong side of the cupboard, which makes the medical records could not be found and the administration staff makes a new medical record. To solve these problems, a web-based management system of patient's medical records was developed for the intention of making the process of managing the patient's medical records easier. The system was developed using Object-Oriented Analysis & Design (OOAD) and Object-Oriented Programming (OOP) methods. The programming language that was used to develop the system is Personal Home Page (PHP) and the framework that was used is Codeigniter. The unit testing and the integration testing were done by using the white-box testing method, while the validation testing was done by using the black-box testing method. The result of unit testing and integration testing has shown that the system runs in accordance with the design that has been made. The result of the validation testing has shown that all the functional requirements have been fulfilled by the system. In other words, the result of all the testing is 100% valid and the system is worth to use.*

**Keywords:** Information System, management of data , medical record, web, object-oriented

## DAFTAR ISI

PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
PRAKATA.....	iv
ABSTRAK.....	v
ABSTRACT .....	vi
DAFTAR ISI .....	vii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR .....	xv
DAFTAR LAMPIRAN .....	xvii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah .....	1
1.3 Tujuan .....	2
1.3.1 Tujuan Umum.....	2
1.3.2 Tujuan Khusus .....	2
1.4 Manfaat.....	2
1.5 Batasan masalah .....	2
1.6 Sistematika pembahasan .....	2
BAB 2 LANDASAN KEPUSTAKAAN .....	4
2.1 Kajian Pustaka .....	4
2.2 Pengelolaan Rekam Medis di Klinik Dr. Syarif .....	5
2.2.1 Pendaftaran Pasien .....	5
2.2.2 Pemeriksaan Pasien .....	6
2.3 Rekayasa Perangkat Lunak .....	6
2.3.1 Model Pengembangan Perangkat Lunak .....	7
2.3.2 Pendekatan Berorientasi Objek .....	11
2.4 Teknologi Pengembangan Sistem.....	14
2.4.1 <i>Codeigniter</i> .....	14
2.4.2 HTML .....	14
2.4.3 PHP .....	15

2.4.4 MySQL .....	15
2.4.5 Javascript.....	15
BAB 3 METODOLOGI PENELITIAN .....	16
3.1 Studi Literatur .....	16
3.2 Kebutuhan.....	16
3.3 Perancangan .....	17
3.4 Implementasi .....	17
3.5 Pengujian .....	18
3.6 Penutup.....	18
BAB 4 KEBUTUHAN.....	19
4.1 Elisitasi dan Analisis Kebutuhan.....	19
4.1.1 Identifikasi Aktor .....	19
4.1.2 Daftar Kebutuhan Sistem .....	19
4.2 Spesifikasi Kebutuhan .....	23
4.2.1 <i>Use Case Diagram</i> .....	24
4.2.2 <i>Use Case Scenario</i> .....	24
BAB 5 PERANCANGAN DAN IMPLEMENTASI .....	41
5.1 Perancangan .....	41
5.1.1 Perancangan Arsitektural.....	41
5.1.2 Perancangan Data .....	46
5.1.3 Perancangan Komponen .....	50
5.1.4 Perancangan Antarmuka.....	51
5.2 Implementasi .....	57
5.2.1 Spesifikasi Sistem .....	57
5.2.2 Batasan Implementasi.....	58
5.2.3 Implementasi Data .....	59
5.2.4 Implementasi Kode Program .....	59
5.2.5 Implementasi Antarmuka .....	60
BAB 6 PENGUJIAN .....	63
6.1 Pengujian Unit.....	63
6.1.1 Pengujian Unit <i>Method</i> tambahPasien() klas Pasien .....	63
6.1.2 Pengujian Unit <i>Method</i> tambahPengguna() klas Pengguna .....	64



6.1.3 Pengujian Unit <i>Method</i> tambahObat() klas Obat .....	65
6.2 Pengujian Integrasi .....	66
6.2.1 Pengujian Integrasi <i>Method</i> tambahPasien() klas PasienController .....	67
6.2.2 Pengujian Integrasi <i>Method</i> tambahPengguna() klas PeggunaController .....	70
6.2.3 Pengujian Integrasi <i>Method</i> tambahObat() Klas ObatController .....	72
6.3 Pengujian Validasi .....	74
6.3.1 Pengujian Validasi <i>Login</i> .....	74
6.3.2 Pengujian Validasi <i>Logout</i> .....	76
6.3.3 Pengujian Validasi Menambah Pegguna .....	77
6.3.4 Pengujian Validasi Melihat Daftar Pegguna .....	78
6.3.5 Pengujian Validasi Mengubah Data Pegguna .....	79
6.3.6 Pengujian Validasi Melihat Daftar Pasien Khusus Petugas Administrasi .....	81
6.3.7 Pengujian Validasi Melihat Daftar Pasien Khusus Dokter .....	82
6.3.8 Pengujian Validasi Menambah Pasien Baru .....	82
6.3.9 Pengujian Validasi Mengubah Data Pasien .....	85
6.3.10 Pengujian Validasi Melihat Rekam Medis .....	87
6.3.11 Pengujian Validasi Melihat Daftar Obat Khusus Dokter .....	87
6.3.12 Pengujian Validasi Melihat Daftar Obat Khusus Apoteker .....	88
6.3.13 Pengujian Validasi Menambah Obat Baru .....	88
6.3.14 Pengujian Validasi Mengubah Data Obat .....	90
6.3.15 Pengujian Validasi Menambah Hasil Pemeriksaan .....	92
6.3.16 Pengujian Validasi Mencetak Rekam Medis .....	93
6.3.17 Pengujian Validasi Melihat Antrian Pasien Khusus Petugas Administrasi .....	94
6.3.18 Pengujian Validasi Menambahkan Pasien Ke dalam Antrian ...	94
6.3.19 Pengujian Validasi Membatalkan Antrian Pasien .....	95
6.3.20 Pengujian Validasi Menambahkan Pasien Ke dalam Ruang Pemeriksaan .....	97
BAB 7 PENUTUP .....	100
7.1 Kesimpulan .....	100
7.2 Saran .....	101

DAFTAR REFERENSI .....	102
LAMPIRAN A HASIL WAWANCARA.....	104

## DAFTAR TABEL

Tabel 2.1 Kajian pustaka .....	4
Tabel 2.2 Simbol pada <i>Use case diagram</i> .....	11
Tabel 2.3 Simbol pada <i>sequence diagram</i> .....	12
Tabel 2.4 Hubungan pada <i>class diagram</i> .....	13
Tabel 4.1 Identifikasi aktor.....	19
Tabel 4.2 Daftar kebutuhan fungsional.....	20
Tabel 4.3 <i>Use case scenario Login</i> .....	25
Tabel 4.4 <i>Use case scenario Logout</i> .....	25
Tabel 4.5 <i>Use case scenario</i> Menambah Pengguna.....	26
Tabel 4.6 <i>Use case scenario</i> Melihat Daftar Pengguna.....	27
Tabel 4.7 <i>Use case scenario</i> Mengubah Data Pengguna .....	27
Tabel 4.8 <i>Use case scenario</i> Melihat Daftar Pasien Khusus Petugas .....	28
Tabel 4.9 <i>Use case scenario</i> Melihat Daftar Pasien Khusus Dokter .....	29
Tabel 4.10 <i>Use case scenario</i> Menambah Pasien Baru.....	30
Tabel 4.11 <i>Use case scenario</i> Mengubah Data Pasien .....	31
Tabel 4.12 <i>Use case scenario</i> Melihat Rekam Medis .....	32
Tabel 4.13 <i>Use case scenario</i> Melihat Daftar Obat Khusus Dokter .....	32
Tabel 4.14 <i>Use case scenario</i> Melihat Daftar Obat Khusus Apoteker .....	33
Tabel 4.15 <i>Use case scenario</i> Menambah Obat Baru .....	34
Tabel 4.16 <i>Use case scenario</i> Mengubah Data Obat .....	35
Tabel 4.17 <i>Use case scenario</i> Menambah Hasil Pemeriksaan .....	35
Tabel 4.18 <i>Use case scenario</i> Mencetak Rekam Medis .....	36
Tabel 4.19. <i>Use case scenario</i> Melihat Antrian Pasien Khusus Petugas Administrasi .....	37
Tabel 4.20. <i>Use case scenario</i> Menambahkan Pasien ke dalam Antrian.....	38
Tabel 4.21. <i>Use case scenario</i> Membatalkan Antrian Pasien .....	38
Tabel 4.22. <i>Use case scenario</i> Menambahkan Pasien Ke dalam Ruang Pemeriksaan .....	39
Tabel 5.1 <i>Pseudocode method</i> tambahPasien() klas Pasien .....	50
Tabel 5.2 <i>Pseudocode method</i> tambahPengguna() klas Pengguna .....	50
Tabel 5.3 <i>Pseudocode method</i> tambahObat() klas Obat .....	50

Tabel 5.4 Penjelasan antarmuka Halaman Menambah Pasien .....	51
Tabel 5.5 Penjelasan antarmuka Halaman Menambah Pengguna .....	54
Tabel 5.6 Penjelasan antarmuka Halaman Menambah Obat Baru .....	56
Tabel 5.7 Spesifikasi perangkat keras .....	57
Tabel 5.8 Spesifikasi perangkat lunak .....	58
Tabel 5.9 Spesifikasi sistem operasi .....	58
Tabel 5.10 Implementasi <i>method</i> tambahPasien() klas Pasien .....	59
Tabel 5.11 Implementasi <i>method</i> tambahPengguna klas Pengguna.....	60
Tabel 5.12 Implementasi <i>method</i> tambahObat() klas Obat .....	60
Tabel 6.1 Hasil pengujian <i>method</i> tambahPasien klas Pasien .....	64
Tabel 6.2 Hasil pengujian <i>method</i> tambahPengguna klas Pengguna .....	65
Tabel 6.3 Hasil pengujian <i>method</i> tambahObat klas Obat .....	66
Tabel 6.4 Hasil pengujian integrasi <i>method</i> tambahPasien() .....	68
Tabel 6.5 Hasil pengujian integrasi <i>method</i> tambahPengguna() .....	71
Tabel 6.6 Hasil pengujian integrasi <i>method</i> tambahObat() .....	73
Tabel 6.7 Pengujian validasi <i>Login</i> .....	74
Tabel 6.8 Pengujian validasi <i>Login</i> alternative satu .....	74
Tabel 6.9 Pengujian validasi <i>Login</i> alternatif dua .....	75
Tabel 6.10 Pengujian validasi <i>Login</i> alternatif tiga .....	75
Tabel 6.11 Pengujian validasi <i>Logout</i> .....	76
Tabel 6.12 Pengujian validasi <i>Logout</i> alternatif satu .....	76
Tabel 6.13 Pengujian validasi Menambah Pengguna .....	77
Tabel 6.14 Pengujian validasi Menambah Pengguna Alternative satu.....	77
Tabel 6.15 Pengujian validasi Menambah Pengguna alternatif dua .....	78
Tabel 6.16 Pengujian validasi Melihat Daftar Pengguna .....	78
Tabel 6.17 Pengujian validasi Mengubah Data Pengguna .....	79
Tabel 6.18 Pengujian validasi Mengubah Data Pengguna alternatif flow satu ....	80
Tabel 6.19 Pengujian validasi Mengubah Data Pengguna alternatif dua .....	80
Tabel 6.20 Pengujian validasi Mengubah Data Pengguna alternative satu.....	81
Tabel 6.21 Pengujian validasi Melihat Daftar Pasien Khusus Petugas Administrasi .....	81
Tabel 6.22 Pengujian validasi Melihat Daftar Pasien Khusus Dokter .....	82
Tabel 6.23 Pengujian validasi Menambah Pasien Baru .....	82

Tabel 6.24 Pengujian validasi Menambah Pasien Baru alternative satu .....	83
Tabel 6.25 Pengujian validasi Menambah Pasien Baru alternatif dua .....	84
Tabel 6.26 Pengujian validasi Menambah Pasien Baru alternatif tiga .....	84
Tabel 6.27 Pengujian validasi Mengubah Data Pasien .....	85
Tabel 6.28 Pengujian validasi Mengubah Data Pasien alternatif satu.....	85
Tabel 6.29 Pengujian validasi Mengubah Data Pasien alternatif dua .....	86
Tabel 6.30 Pengujian validasi Mengubah Data Pasien alternatif tiga .....	86
Tabel 6.31 Pengujian validasi Melihat Rekam Medis.....	87
Tabel 6.32 Pengujian validasi Melihat Daftar Obat Khusus Dokter .....	87
Tabel 6.33 Pengujian validasi Melihat Daftar Obat Khusus Apoteker .....	88
Tabel 6.34 Pengujian validasi Menambah Obat Baru .....	88
Tabel 6.35 Pengujian validasi Menambah Obat Baru alternatif satu .....	89
Tabel 6.36 Pengujian validasi Menambah Obat Baru alternative dua .....	89
Tabel 6.37 Pengujian validasi Menambah Obat Baru alternatif tiga .....	90
Tabel 6.38 Pengujian validasi Mengubah Data Obat .....	90
Tabel 6.39 Pengujian validasi Mengubah Data Obat alternatif satu .....	91
Tabel 6.40 Pengujian validasi Mengubah Data Obat alternatif dua .....	92
Tabel 6.41 Pengujian validasi Menambah Hasil Pemeriksaan.....	92
Tabel 6.42 Pengujian validasi Menambah Hasil Pemeriksaan alternatif dua.....	93
Tabel 6.43 Pengujian validasi Mencetak Rekam Medis.....	93
Tabel 6.44 Pengujian validasi Melihat Antrian Pasien Khusus Petugas Administrasi .....	94
Tabel 6.45 Pengujian validasi Menambahkan Pasien Ke dalam Antrian .....	94
Tabel 6.46 Pengujian validasi Menambahkan Pasien Ke dalam Antrian alternatif satu.....	95
Tabel 6.47 Pengujian validasi Membatalkan Antrian Pasien .....	96
Tabel 6.48 Pengujian validasi Membatalkan Antrian Pasien alternatif satu .....	96
Tabel 6.49 Pengujian validasi Menambahkan Pasien ke dalam Ruang Pemeriksaan .....	97
Tabel 6.50 Pengujian validasi Menambahkan Pasien ke dalam Ruang Pemeriksaan alternatif satu.....	97
Tabel 6.51 Pengujian validasi Menambahkan Pasien ke dalam Ruang Pemeriksaan alternatif dua.....	98

Tabel 6.52 Pengujian validasi Menambahkan Pasien ke dalam Ruang Pemeriksaan alternatif tiga.....	99
--	----

## DAFTAR GAMBAR

Gambar 2.1 Proses bisnis pendaftaran pasien (saat ini).....	5
Gambar 2.2 Proses bisnis pemeriksaan pasien (saat ini) .....	6
Gambar 2.3 Model <i>Waterfall</i> .....	7
Gambar 3.1 Diagram alir metodologi penelitian .....	16
Gambar 4.1 <i>Use case diagram</i> Sistem Pengelolaan Rekam Medis Pasien Berbasis Web .....	24
Gambar 5.1 <i>Sequence diagram</i> Menambah Pengguna .....	42
Gambar 5.2 <i>Sequence diagram</i> Menambah Pasien Baru .....	43
Gambar 5.3 <i>Sequence diagram</i> Menambah Obat Baru .....	44
Gambar 5.4 Pemodelan <i>class diagram</i> umum.....	45
Gambar 5.5 Pemodelan <i>class diagram controller</i> .....	45
Gambar 5.6 Pemodelan <i>class diagram entity</i> .....	46
Gambar 5.7 Pemodelan CDM .....	47
Gambar 5.8 Detail entitas pengguna .....	47
Gambar 5.9 Detail entitas pekerjaan .....	47
Gambar 5.10 Detail entitas obat.....	48
Gambar 5.11 Detail entitas pasien.....	48
Gambar 5.12 Detail entitas antrian.....	48
Gambar 5.13 Detail entitas ruangan .....	48
Gambar 5.14 Detail entitas data_medis .....	49
Gambar 5.15 Pemodelan PDM.....	49
Gambar 5.16 Perancangan antarmuka Halaman Menambah Pasien.....	51
Gambar 5.17 Perancangan antarmuka Halaman Menambah Pengguna .....	54
Gambar 5.18 Perancangan antarmuka Halaman Menambah Obat Baru.....	55
Gambar 5.19 Implementasi data .....	59
Gambar 5.20 Implementasi antarmuka Halaman Menambah Pasien .....	61
Gambar 5.21 Implementasi antarmuka Halaman Menambah Pengguna .....	61
Gambar 5.22 Implementasi antarmuka Halaman Menambah Obat Baru.....	62
Gambar 6.1 <i>Flow graph method tambahpasien()</i> .....	63
Gambar 6.2 <i>Flow graph method tambahPengguna()</i> .....	64
Gambar 6.3 <i>Flow graph method tambahObat()</i> .....	66

Gambar 6.4 <i>Flow graph</i> pengujian integrasi <i>method</i> tambahPasien() klas PasienController.....	68
Gambar 6.5 <i>Flow graph</i> pengujian integrasi <i>method</i> tambahPengguna() klas PenggunaController .....	70
Gambar 6.6 <i>Flow graph</i> pengujian integrasi <i>method</i> tambahObat() klas ObatController .....	72



## **DAFTAR LAMPIRAN**

LAMPIRAN A HASIL WAWANCARA.....	104
---------------------------------	-----

# **BAB 1 PENDAHULUAN**

## **1.1 Latar Belakang**

Klinik merupakan fasilitas pelayanan kesehatan yang melakukan kegiatan pada bidang pelayanan medis dasar dan/atau spesialis. Pelayanan medis dasar tersebut diselenggarakan oleh lebih dari satu jenis tenaga kesehatan dan dipimpin oleh seorang tenaga medis. Klinik harus mampu memberikan pelayanan yang aman dan bermutu dengan mengutamakan kepentingan terbaik pasien sesuai dengan standar profesi.

Klinik Dr. Syarif merupakan salah satu klinik pratama yang berada di Kota Samarinda, Kalimantan Timur. Sama seperti klinik pada umumnya, Klinik Dr. Syarif memiliki rekam medis pasien yang telah berobat pada klinik tersebut dan obat untuk mengobati pasien tersebut. Klinik tersebut memiliki beberapa macam bidang pekerjaan, seperti petugas administrasi, dokter, dan apoteker. Petugas administrasi melakukan pendataan terhadap pasien yang datang untuk berobat dan membuat rekam medis untuk pasien baru. Petugas administrasi juga bertugas untuk menyiapkan rekam medis pasien bagi pasien yang telah berobat pada klinik tersebut. Dokter melakukan pemeriksaan terhadap pasien, dan melakukan penambahan informasi pada rekam medis pasien terkait hasil dari pemeriksaan yang dilakukan sebelumnya. Apoteker bertugas dalam melakukan pengelolaan obat, meraciknya berdasarkan resep dokter dan menjualnya kepada pasien.

Pada Klinik Dr. Syarif, proses pencatatan rekam medis masih dilakukan dengan cara menulis. Hal ini dapat menyebabkan masalah, yaitu terjadinya duplikasi rekam medis dikarenakan pasien yang lupa jika pernah berobat pada klinik tersebut. Pasien yang lupa tersebut akhirnya memilih untuk membuat rekam medis baru. Proses pencarian rekam medis juga tergolong sulit, dikarenakan jumlah rekam medis yang sangat banyak pada tempat penyimpanan.

Berdasarkan permasalahan tersebut, penelitian dilakukan dengan judul "Pengembangan Sistem Pengelolaan Rekam Medis Pasien Berbasis Web. Studi Kasus: Klinik Dr. Syarif". Sistem tersebut diharapkan dapat membantu orang-orang yang bekerja pada Klinik Dr. Syarif dalam pengelolaan rekam medis pasien.

## **1.2 Rumusan Masalah**

Rumusan masalah yang dapat dijadikan sebagai bahan skripsi ini berdasarkan latar belakang yang telah dijelaskan sebelumnya ialah:

1. Apa kebutuhan dari sistem pengelolaan rekam medis pasien berbasis web di klinik Dr. Syarif?
2. Bagaimana hasil dari perancangan dan implementasi sistem pengelolaan rekam medis pasien berbasis web?

3. Bagaimana hasil dari pengujian sistem pengelolaan rekam medis pasien berbasis web?

### **1.3 Tujuan**

Tujuan dari penelitian ini adalah:

#### **1.3.1 Tujuan Umum**

Mengembangkan sistem pengelolaan rekam medis pasien berbasis web.

#### **1.3.2 Tujuan Khusus**

1. Merekayasa kebutuhan untuk membangun sistem pengelolaan rekam medis pasien berbasis web.
2. Merancang dan membuat sistem pengelolaan rekam medis pasien berbasis web.
3. Menguji semua kebutuhan dari sistem pengelolaan rekam medis pasien berbasis web.

### **1.4 Manfaat**

Manfaat dari penelitian ini adalah:

1. Dapat membantu petugas administrasi dalam mengelola rekam medis pasien pada klinik Dr. Syarif.
2. Dapat membantu dokter untuk mendapatkan informasi mengenai stok obat yang terdapat di klinik.

### **1.5 Batasan Masalah**

Batasan masalah penelitian ini adalah:

1. Objek penelitian dilakukan di klinik Dr. Syarif.
2. Bahasa pemrograman yang digunakan adalah PHP dan basis data MySQL serta menggunakan *framework Codeigniter*.
3. Sistem berbasis web.
4. Pemodelan kebutuhan menggunakan pendekatan berorientasi objek.
5. Sistem digunakan oleh admin, petugas administrasi, apoteker, dan dokter.

### **1.6 Sistematika pembahasan**

Sistematika pembahasan dari skripsi ini adalah:

#### **BAB 1 PENDAHULUAN**

Bab ini akan membahas mengenai latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika pembahasan dalam penelitian pengembangan sistem pengelolaan rekam medis pasien berbasis web.

## **BAB 2 LANDASAN KEPUSTAKAAN**

Bab ini akan menguraikan tentang kajian pustaka dan dasar-dasar teori yang berhubungan dengan penelitian ini. Kajian pustaka berisi informasi mengenai penelitian-penelitian terdahulu yang berkaitan dengan penelitian ini. Dasar teori berisi informasi mengenai domain masalah dan rekayasa perangkat lunak, serta membahas teknologi-teknologi yang digunakan dalam pengembangan sistem yaitu Codeigniter, HTML, PHP, MySQL, dan Javascript.

## **BAB 3 METODOLOGI PENELITIAN**

Bab ini akan menguraikan tentang tahapan-tahapan yang dilakukan pada saat penelitian yang terdiri dari studi literatur, kebutuhan, perancangan, implementasi, pengujian, dan kesimpulan dalam penelitian sistem informasi.

## **BAB 4 KEBUTUHAN**

Bab ini menguraikan tentang kebutuhan sistem mulai dari proses yang dilakukan dalam penggalan kebutuhan, kebutuhan fungsional, spesifikasi kebutuhan, *use case diagram* dan *use case scenario*.

## **BAB 5 PERANCANGAN DAN IMPLEMENTASI**

Bab ini akan membahas mengenai perancangan dari pengembangan sistem pengelolaan rekam medis pasien berbasis web. Perancangan memuat pemodelan *sequence diagram*, *class diagram*, perancangan data, perancangan komponen, dan perancangan antarmuka. Bab ini juga akan membahas mengenai implementasi yang dilakukan pada pengembangan sistem pengelolaan rekam medis pasien berbasis web. Implementasi memuat spesifikasi sistem, batasan implementasi, implementasi data, implementasi kode program, dan implementasi antarmuka.

## **BAB 6 PENGUJIAN**

Bab ini akan membahas mengenai pengujian yang dilakukan pada pengembangan sistem pengelolaan rekam medis pasien berbasis web. Pengujian terdiri dari pengujian unit, pengujian integrasi, dan pengujian validasi.

## **BAB 7 PENUTUP**

Bab ini akan membahas mengenai kesimpulan dan saran dari penelitian ini. Kesimpulan diperoleh dari hasil tahap kebutuhan, perancangan sistem, implementasi sistem, dan pengujian. Saran berisi informasi yang dapat digunakan untuk proses pengembangan lanjut sistem.

## BAB 2 LANDASAN KEPUSTAKAAN

### 2.1 Kajian Pustaka

Sebuah penelitian telah dilakukan oleh Widya dan Migunani (2014) yang menjelaskan mengenai bagaimana membuat sistem informasi berbasis web. Pengembangan sistem pada penelitian tersebut menggunakan SDLC dengan metode *waterfall* dan pendekatan *object oriented analysis and design*. Hasil dari penelitian tersebut adalah menghasilkan aplikasi data rekam medis pasien berbasis web dan mengurangi kemungkinan kehilangan data rekam medis pasien yang disebabkan oleh penyimpanan data di dalam almari.

Penelitian lainnya telah dilakukan oleh Ayuliana (2014) yang menjelaskan mengenai pembangunan sistem informasi dalam penyimpanan dan pengolahan data operasional. Pengembangan sistem pada penelitian tersebut menggunakan SDLC dengan metode *waterfall* dan pendekan objek terstruktur. Hasil dari penelitian tersebut adalah informasi dan data yang terdapat pada sistem dapat diintegrasikan dengan baik, cepat, dan akurat. Sistem juga dapat memberikan bantuan dalam meningkatkan kualitas pelayanan klinik dan mempermudah pengguna melakukan penyimpanan dan pengaksesan data. Tabel kajian pustaka dapat dilihat pada Tabel 2.1.

**Tabel 2.1 Kajian pustaka**

No	Judul Penelitian	Objek Penelitian	Perbedaan	
			Penelitian Terdahulu	Rencana Penelitian
1	Rancang Bangun Sistem Informasi Rekam Medis Berbasis Web (Studi Kasus: Rumah Perawatan Psiko-Neuro-Geriatri “Puri Saras” Semarang) (Widya & Migunani, 2014)	Pembangunan sistem informasi rekam medis berbasis web.	Hanya terdapat dua aktor, yaitu petugas administrasi dan dokter.	Menambah jenis aktor yaitu admin dan apoteker untuk meningkatkan efisiensi sistem.
2	Perancangan Sistem Informasi Manajemen Pada Klinik Skala Kecil (Studi Kasus: Klinik Dr. Jonni) (Ayuliana, 2014)	Pembangunan sistem informasi manajemen pada klinik skala kecil.	Belum ada fitur antrian pasien.	Menambah fitur antrian pasien untuk meningkatkan efisiensi sistem.

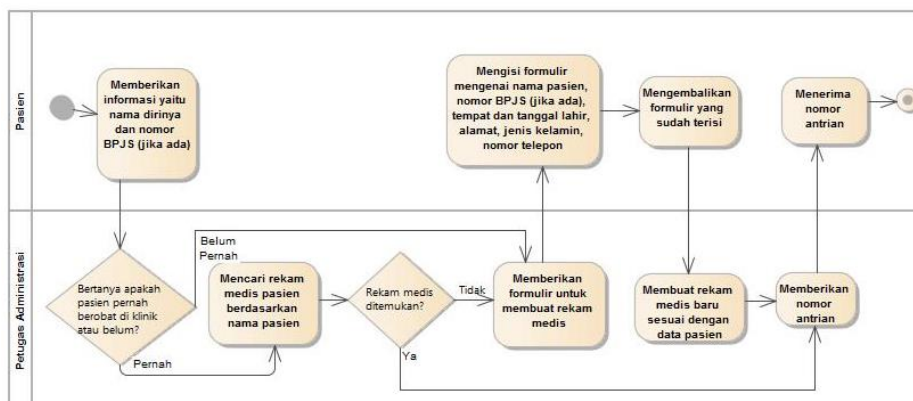
## 2.2 Pengelolaan Rekam Medis di Klinik Dr. Syarif

Menurut Huffman (1994), rekam medis merupakan salah satu bentuk dari berkas yang menyatakan siapa pasien yang dirawat, apa keluhan yang diutarakan oleh pasien, dan mengapa pasien bisa mendapatkan keluhan tersebut. Rekam medis juga berisi tentang dimana pemeriksaan kesehatan pasien dilakukan, kapan pemeriksaan tersebut dilaksanakan, dan bagaimana pelayanan yang diperoleh seorang pasien. Pengelolaan rekam medis dilaksanakan sesuai dengan organisasi dan tata kerja sarana pelayanan kesehatan (Permenkes RI No. 269, 2008).

Berdasarkan hasil wawancara dengan Dr. Syarif (Lampiran A), rekam medis pada klinik Dr. Syarif berisi data diri pasien, yaitu nama, nomor BPJS jika ada, tempat & tanggal lahir, dan nomor telepon. Selain itu, rekam medis pada klinik Dr. Syarif juga memuat tanggal pemeriksaan pasien, dokter yang melakukan pemeriksaan, keluhan pasien, hasil pemeriksaan fisik, diagnosa, obat yang dibutuhkan, dan catatan tambahan. Rekam medis tersebut ditulis pada kertas dan disimpan di dalam almari.

### 2.2.1 Pendaftaran Pasien

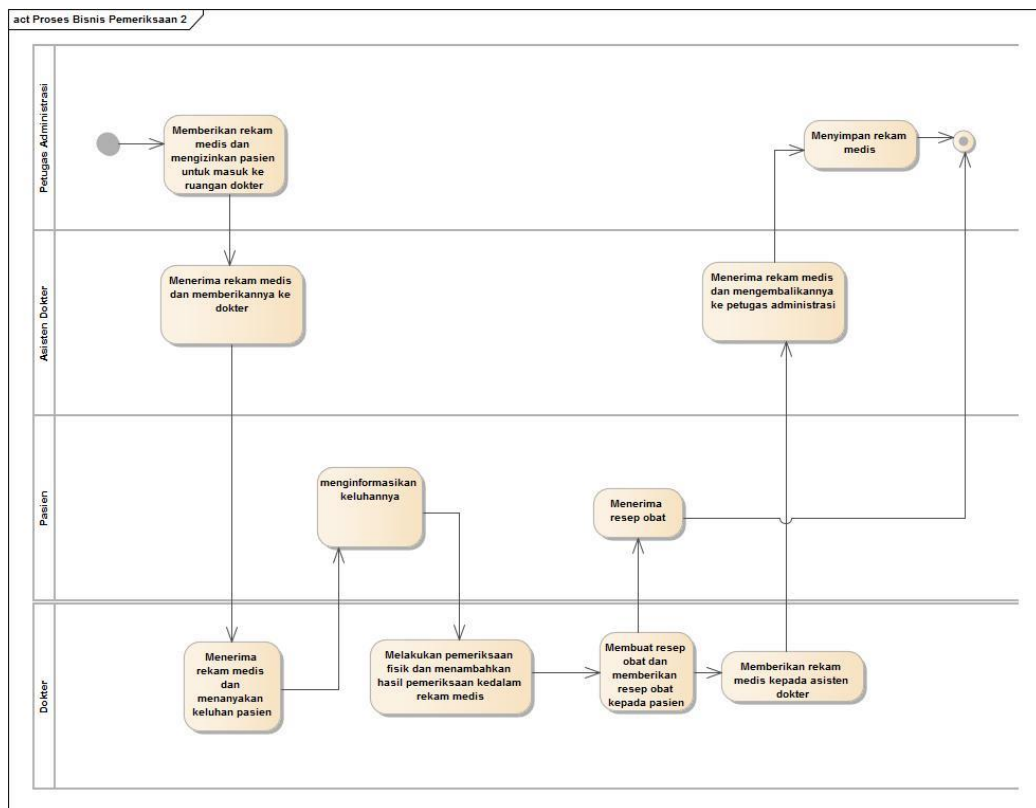
Berdasarkan hasil wawancara dengan Dr. Syarif (Lampiran A), pasien memberitahukan nama dirinya dan nomor BPJS (jika ada) kepada petugas administrasi. Petugas administrasi kemudian bertanya kepada pasien jika pasien tersebut sudah pernah berobat sebelumnya di klinik atau belum. Jika pernah, petugas akan mencari rekam medis di tempat penyimpanan yang terdapat pada klinik. Jika belum pernah, petugas akan memberikan formulir untuk keperluan pembuatan rekam medis pasien yang terdiri dari nama pasien, nomor BPJS jika ada, tempat dan tanggal lahir, alamat, jenis kelamin, dan nomor telepon. Pasien mengisi formulir secara lengkap dan mengembalikannya kepada petugas administrasi. Petugas administrasi lalu membuat rekam medis pasien sesuai formulir tersebut. Petugas kemudian memberikan nomor antrian kepada pasien. Proses bisnis pendaftaran pasien dapat dilihat pada Gambar 2.1.



**Gambar 2.1** Proses bisnis pendaftaran pasien (saat ini)

### 2.2.2 Pemeriksaan Pasien

Berdasarkan hasil wawancara dengan Dr. Syarif (Lampiran A), jika antrian sudah menjadi giliran pasien, maka asisten dokter membawa rekam medis pasien dan memasuki ruangan pemeriksaan bersama pasien. Asisten dokter kemudian memberikan rekam medis tersebut kepada dokter. Pasien kemudian memberitahukan keluhannya. Dokter memeriksa fisik pasien lalu menambahkan hasil pemeriksaan ke dalam rekam medis. Hasil pemeriksaan tersebut terdiri dari tanggal pemeriksaan, keluhan pasien, hasil pemeriksaan fisik, diagnosa, obat yang dibutuhkan, dan catatan tambahan. Dokter kemudian membuat resep obat lalu memberikannya kepada pasien. Dokter memberikan rekam medis pasien kepada asisten dokter. Pasien menerima resep obat dan keluar ruangan. Asisten dokter menerima rekam medis dan mengembalikan rekam medis kepada petugas administrasi. Proses bisnis pemeriksaan pasien dapat dilihat pada Gambar 2.2.



**Gambar 2.2 Proses bisnis pemeriksaan pasien (saat ini)**

### 2.3 Rekayasa Perangkat Lunak

Rekayasa Perangkat Lunak merupakan suatu proses yang dilakukan dalam upaya pembuatan dan penggunaan prinsip-prinsip keahlian teknik. Prinsip-prinsip keahlian teknik tersebut berguna untuk mendapatkan perangkat lunak yang handal dan efisien dalam bekerja pada mesin yang sesungguhnya (Pressman, 2010).

Menurut Roger S. Pressman (2010), perangkat lunak berbeda dengan program komputer. *Software* bukan hanya mencakup program, melainkan juga semua dokumentasi dan konfigurasi data yang berhubungan dalam upaya untuk membuat program yang dapat beroperasi dengan benar. Menurut Pressman (2010), *software* merupakan kumpulan dari beberapa instruksi yang akan menyediakan fungsi yang diperlukan ketika dijalankan.

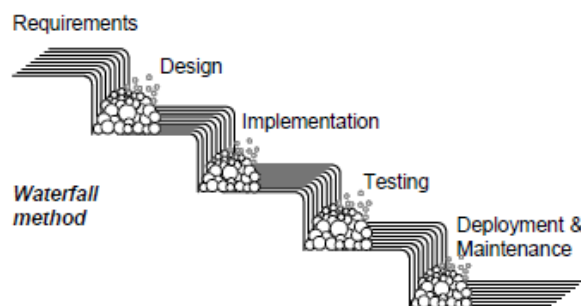
Tujuan dari dilakukannya rekayasa perangkat lunak adalah untuk menghasilkan perangkat lunak yang memiliki kinerja yang tinggi, handal, efisien, dan tepat waktu. Tujuan lain dari dilakukannya rekayasa perangkat lunak ialah mengurangi biaya produksi perangkat lunak, biaya perawatan murah, dan terjangkau. Model pengembangan perangkat lunak digunakan untuk membantu proses pengembangan perangkat lunak.

### 2.3.1 Model Pengembangan Perangkat Lunak

Model pengembangan perangkat lunak merupakan suatu strategi pengembangan perangkat lunak yang bertujuan untuk menghasilkan perangkat lunak yang berkualitas. Model-model dalam pengembangan perangkat lunak menjelaskan mengenai teknik untuk membangun perangkat lunak, seperti rekayasa kebutuhan, perancangan, implementasi, pengujian, dan pemeliharaan. *System Development Life-Cycle* (SDLC) merupakan istilah yang dapat digunakan untuk merujuk kepada model pengembangan perangkat lunak. SDLC memiliki beberapa model, seperti model *waterfall*, model *prototype*, model *evolutionary development*, model *agile*, dan model *Rapid Application Development* (RAD). Penelitian ini menggunakan model *waterfall* dikarenakan tidak adanya perubahan dalam kebutuhan sistem.

#### 2.3.1.1 Model *Waterfall*

Model *waterfall* atau yang biasa disebut model sekuensial linear merupakan model dalam proses pengembangan sistem yang sistematis. Tahapan-tahapan yang terdapat pada model ini dilakukan secara berurutan. Menurut Marsic (2012), beberapa tahapan tersebut adalah: *Requirements*, *Design*, *Implementation*, *Testing*, dan *Deployment and Maintenance*. Suatu tahapan harus diselesaikan terlebih dahulu sebelum melakukan tahapan berikutnya. Tahapan model *waterfall* dapat dilihat pada Gambar 2.3.



**Gambar 2.3 Model *Waterfall***

Sumber: Marsic (2012)



Penjelasan tahapan model *waterfall* menurut Marsic (2012) ialah:

1. *Requirements* (Kebutuhan)

Tahapan kebutuhan merupakan tahapan pendokumentasian *use case* dan menurunkan model domain statis. Tahap kebutuhan terdiri dari tiga tahapan, yaitu:

a) Pengumpulan kebutuhan (*requirements gathering*)

Pengumpulan kebutuhan (dikenal juga sebagai elisitasi kebutuhan) membantu pelanggan untuk menentukan apa yang dibutuhkan: apa yang harus dicapai, bagaimana sistem akan memenuhi kebutuhan, dan bagaimana sistem akan digunakan dalam kegiatan sehari-hari.

b) Analisis kebutuhan (*requirements analysis*)

Analisis kebutuhan melibatkan penyaringan dan modifikasi kebutuhan yang diperoleh dari pelanggan selama proses pengumpulan kebutuhan. Diskusi dengan pelanggan diperlukan untuk menentukan prioritas, apa yang penting, siapa yang melakukan, dan kapan itu dibutuhkan. Penting untuk memastikan bahwa pemahaman *developer* akan masalah memiliki kesamaan dengan pemahaman pelanggan akan masalah.

c) Spesifikasi kebutuhan (*requirements specification*)

Spesifikasi kebutuhan merepresentasikan masalah ke dalam suatu cara informal maupun formal untuk memastikan kejelasan, konsistensi, dan kelengkapan. Suatu spesifikasi bisa menjadi dokumen tertulis, suatu kelompok model grafik, suatu *use case*, suatu *prototype*, ataupun kombinasi dari hal-hal tersebut.

2. *Design* (Perancangan)

Tahap perancangan merupakan tahap pemberian tugas kepada komponen-komponen dan melakukan spesifikasi dinamis dari interaksi komponen-komponen tersebut berdasarkan *use case scenario* yang berbeda. Tahap perancangan berurusan dengan bagaimana elemen dari sistem bekerja dan berinteraksi. Menurut Sommerville (2011), tahap ini terbagi menjadi beberapa bagian, yaitu:

a) Perancangan Arsitektural

Perancangan arsitektural adalah perancangan yang mendefinisikan elemen-elemen utama atau sub-sistem dan relasi antarelemen tersebut di dalam sistem.

b) Perancangan Data

Perancangan data bertujuan untuk mentransformasikan model domain informasi dari hasil analisis menjadi struktur data yang dibutuhkan dalam tahap implementasi sistem.

c) Perancangan Komponen

Perancangan komponen merupakan tahap dekomposisi sub-sistem menjadi komponen detail. Setiap komponen sistem dirancang mengenai bagaimana komponen-komponen tersebut bekerja.

d) Perancangan Antarmuka

Perancangan antarmuka bertujuan untuk membuat antarmuka di antara komponen sistem. Spesifikasi antarmuka harus tidak ambigu. Dengan antarmuka yang tepat, suatu komponen dapat digunakan tanpa komponen lain mengetahui bagaimana komponen diimplementasikan. Setelah spesifikasi antarmuka disetujui, komponen dapat dirancang dan dikembangkan secara bersamaan.

3. *Implementation* (Implementasi)

Tahap implementasi merupakan tahapan penerjemahan hasil dari tahap perancangan menjadi kode program atau bentuk bahasa yang dapat dibaca oleh mesin. Proses implementasi dapat dilakukan dengan menggunakan bahasa pemrograman seperti *Hypertext Markup Language* (HTML), *Javascript*, dan *Hypertext Preprocessor* (PHP). Hasil akhir dari tahap implementasi adalah kode program (Kumar, 2013).

4. *Testing* (Pengujian)

Tahap pengujian dilakukan untuk menguji unit yang terdapat pada sistem dan menguji keseluruhan sistem. Pengujian keseluruhan sistem terdiri dari pengujian integrasi dan pengujian validasi. Menurut Pressman (2010), pengujian dimulai dengan melakukan pengujian unit, dilanjutkan dengan pengujian integrasi, dan diakhiri dengan pengujian validasi. Pengujian unit menitikberatkan pada proses verifikasi unit terkecil pada rancangan perangkat lunak, yaitu komponen atau modul (Pressman, 2010). Pengujian integrasi merupakan tahapan pengujian komposisi dari komponen sistem. Pengujian integrasi memiliki beberapa pendekatan, antara lain *top-down integration* dan *bottom-up integration* (Pressman, 2010). Menurut Pressman (2010), *top-down integration* merupakan pengujian yang dimulai pada komponen dengan tingkat tertinggi menuju komponen tingkat terendah. *Bottom-up integration* merupakan pengujian yang dilakukan mulai dari komponen dengan tingkat terendah pada struktur program menuju komponen tingkat tertinggi. Pengujian validasi merupakan pengujian dimana kebutuhan yang telah ditetapkan sebagai bagian dari pemodelan kebutuhan divalidasi terhadap sistem yang telah dibangun (Pressman, 2010).

Menurut Pressman (2010), sistem dapat diuji melalui dua perspektif berbeda, yaitu menguji logika internal program dan menguji kebutuhan sistem. Pengujian logika internal program disebut dengan *white box testing* dan pengujian kebutuhan sistem disebut dengan *black box testing*. Pengujian unit dan integrasi menerapkan teknik *white box testing* dan pengujian validasi menerapkan teknik *black box testing*. Penjelasan *white box testing* dan *black box testing* menurut Pressman (2010) ialah:

a) *White Box Testing*

*White box testing* merupakan teknik untuk melakukan pengujian pada struktur kontrol sistem. Tujuan dari *white box testing* ialah:

- Memastikan seluruh jalur independen telah dilewati setidaknya satu kali.
- Melaksanakan seluruh keputusan logika pada sisi benar dan salah.
- Menjalankan semua *loop* di dalam batasan operasional.
- Melatih struktur data internal untuk memastikan keabsahannya.

Pengujian *white box* dapat dilakukan dengan dua metode, yaitu *basis path testing* dan *control structure testing*. *Basis path testing* merupakan pengujian yang memungkinkan perancangan *test case* (kasus uji) untuk menurunkan ukuran kompleksitas logika dan menggunakan ukuran tersebut sebagai petunjuk untuk menentukan rangkaian dasar jalur eksekusi. Tahap-tahap dari pengujian *basis path* adalah:

- 1) Membuat *flow graph* berdasarkan pemetaan diagram alir.
- 2) Menghitung *cyclomatic complexity*, dengan cara:
  - a. menghitung wilayah (*region*) yang dibatasi oleh *edge* dan *node*.
  - b. Menghitung jumlah  $edge - node + 2$ .
  - c. Menghitung *Node* yang bercabang (*Predicate node*) +1.
- 3) Menentukan *Independent path* dari hasil *flow graph*.
- 4) Membuat kasus uji berdasarkan *independent path* yang telah diketahui sebelumnya.

b) *Black Box Testing*

*Black box testing* atau *behavioral testing* berfokus kepada kebutuhan fungsional perangkat lunak. Pengujian *black box* memungkinkan untuk membuat kumpulan kondisi masukan yang akan melatih seluruh kebutuhan fungsional dari suatu program. Salah satu metode pengujian *black box* adalah *scenario-based testing*. *Scenario-based testing* berfokus pada apa yang *user* lakukan, bukan pada apa yang produk lakukan. *Scenario-based testing* akan mengungkap kesalahan yang terjadi ketika aktor berinteraksi dengan sistem. *Scenario-based testing* dilakukan dengan membuat kasus uji berdasarkan skenario yang ada pada *use case scenario*.

5. *Deployment and Maintenance* (Penyebaran dan Perbaikan)

Tahapan penyebaran dan perbaikan merupakan tahap dimana sistem dijalankan pada lingkungan yang nyata, kesalahan (*bug*) diperbaiki, dan fitur-fitur baru ditambahkan.

### 2.3.2 Pendekatan Berorientasi Objek

Menurut Satzinger *et al* (2010), pendekatan berorientasi objek ialah pendekatan untuk mengembangkan sistem yang memandang bahwa sistem sebagai sekumpulan objek yang bekerjasama dan saling berinteraksi dalam upaya menyelesaikan tugas-tugasnya. Pendekatan berorientasi objek menurut Booch (2007) terdiri dari:

1. *Object-Oriented Analysis (OOA)* merupakan metode analisis yang mengidentifikasi kebutuhan melalui perspektif dari kelas-kelas atau objek-objek yang terdapat pada domain masalah.
2. *Object-Oriented Design (OOD)* merupakan metode perancangan yang meliputi proses dekomposisi berorientasi objek dan suatu notasi untuk menggambarkan model logika dan fisik serta model statis dan dinamis dari sistem.
3. *Object Oriented Programming (OOP)* merupakan suatu metode implementasi dimana program diatur sebagai sekumpulan objek yang saling bekerja sama, dimana setiap objek merepresentasikan suatu instansiasi dari suatu kelas, dan kelas-kelas tersebut saling terhubung melalui hubungan pewarisan.

Proses pemodelan berorientasi objek dapat dilakukan dengan menggunakan *Unified Modelling Language (UML)* (Sommerville, 2011).



#### 2.3.2.1 Unified Modelling Language (UML)


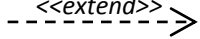
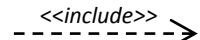
Menurut Sommerville (2011), UML merupakan suatu bahasa grafis yang digunakan di dalam pengembangan berorientasi objek. UML terdiri dari beberapa jenis model sistem. Beberapa jenis model sistem tersebut menyediakan berbagai pandangan yang berbeda tentang suatu sistem. Jenis-jenis model sistem tersebut antara lain *use case diagram*, *sequence diagram*, dan *class diagram*.

##### 1. Use Case Diagram

*Use case diagram* merupakan diagram yang menunjukkan interaksi antara sistem dan aktor eksternal (Sommerville, 2011). Simbol-simbol pada *use case diagram* menurut Rumbaugh (2004) dapat dilihat pada Tabel 2.2.

**Tabel 2.2 Simbol pada *Use case diagram***

No	Simbol>Nama	Deskripsi
1	Aktor 	Sebuah idealisasi peran yang dimainkan oleh orang, proses, atau benda yang berinteraksi dengan sistem, subsistem, atau kelas.
2	Use case 	Fungsionalitas yang nampak dari luar, yang disediakan oleh sistem dengan tujuan untuk mendefinisikan perilaku tanpa mengungkapkan struktur internal sistem.


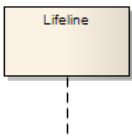

3	Asosiasi 	Jalur komunikasi antara seorang aktor dan sebuah <i>use case</i> yang berpartisipasi dalam komunikasi tersebut.
4	<i>Extend</i> 	Hubungan dari sebuah <i>extension use case</i> menuju <i>base use case</i> , menspesifikasikan bagaimana perilaku pada <i>extension use case</i> dapat dimasukkan ke dalam perilaku dari <i>base use case</i> .
5	<i>Include</i> 	Hubungan dari sebuah <i>base use case</i> menuju sebuah <i>inclusion use case</i> ( <i>use case</i> yang disertakan), menspesifikasikan bahwa perilaku yang terdapat pada <i>inclusion use case</i> akan dimasukkan ke dalam perilaku yang terdapat pada <i>base use case</i> .





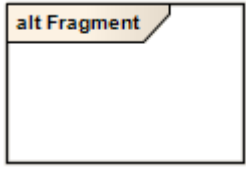
Sumber: Rumbaugh (2004)

## 2. Sequence Diagram

Menurut Sommerville (2011), *sequence diagram* merupakan suatu diagram yang menunjukkan interaksi antara aktor dan sistem, dan objek-objek di dalam sistem. Menurut Fowler (2003), *sequence diagram* menampilkan beberapa objek dan pesan-pesan yang dikirimkan di antara objek-objek tersebut dalam suatu *use case*. Simbol-simbol pada *sequence diagram* menurut Sparx Systems (1996) dapat dilihat pada Tabel 2.3.

**Tabel 2.3 Simbol pada *sequence diagram***

No	Simbol>Nama	Deskripsi
1	Aktor 	<i>Actor</i> merupakan pengguna sistem. Pengguna bisa diartikan sebagai manusia, mesin, atau sistem lain. Segala yang berinteraksi dengan sistem melalui luar sistem atau <i>boundary</i> sistem dianggap <i>actor</i> .
2	<i>Lifeline</i> 	<i>Lifeline</i> merepresentasikan suatu elemen terhubung yang berbeda dan merupakan suatu peserta individu dalam suatu interaksi.
3	<i>Boundary</i> 	Suatu objek yang memodelkan batasan sistem, khususnya <i>user interface</i> . Suatu <i>boundary</i> digunakan untuk menangkap interaksi pengguna dengan sistem.


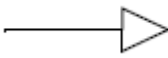
4	<i>Control</i> 	Suatu objek yang memodelkan suatu entitas pengontrol atau pengelola.
5	<i>Entity</i> 	Suatu objek yang memodelkan suatu penyimpanan atau mekanisme yang menangkap informasi atau pengetahuan di dalam sistem.
6	<i>Message</i> 	<i>Message</i> mengindikasikan suatu aliran informasi atau transisi kontrol di antara elemen.
7	<i>Self-Message</i> 	<i>Self-message</i> mencerminkan proses baru yang dipanggil di dalam <i>lifeline</i> pemanggil.
8	<i>Fragment</i> 	<i>Fragment</i> merepresentasikan iterasi atau proses alternatif di dalam <i>sequence diagram</i> .


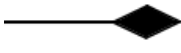
Sumber: Sparx Systems (1996)

### 3. Class Diagram

Menurut Sommerville (2011), *class diagram* merupakan suatu tipe diagram UML yang menunjukkan klas-klas objek di dalam sebuah sistem dan hubungan antarklas tersebut. Objek merepresentasikan suatu benda di dunia nyata, seperti pasien, resep, dan dokter. Suatu objek memiliki informasi berupa nama, atribut (karakteristik dari objek), dan operasi (sesuatu yang bisa dilakukan oleh suatu objek). Beberapa hubungan antarklas menurut Sparx Systems (1996) dapat dilihat pada Tabel 2.4.

**Tabel 2.4 Hubungan pada *class diagram***

Nama hubungan	Simbol	Keterangan
Asosiasi		Hubungan asosiasi mengartikan bahwa dua model elemen mempunyai suatu relasi, biasanya diimplementasikan sebagai suatu variabel instans di dalam satu atau kedua klas.
Generalisasi		Sebuah relasi yang digunakan untuk mengindikasikan suatu pewarisan. Relasi dihubungkan dari klas spesifik menuju klas umum, yang menandakan klas spesifik mewarisi karakteristik klas umum.

Agregasi		Sebuah relasi yang menandakan suatu elemen memuat atau tersusun dari elemen lain.
Komposisi		Sebuah relasi yang menggambarkan suatu elemen terbuat dari komponen-komponen yang lebih kecil. Jika suatu komposisi dihapus, seluruh bagiannya juga dapat terhapus. Namun, suatu bagian dapat dihapus secara individu dari suatu komposisi tanpa harus menghapus keseluruhan komposisi.

Sumber: Sparx Systems (1996)

## 2.4 Teknologi Pengembangan Sistem

Pengembangan sistem pengelolaan rekam medis pasien berbasis *web* memerlukan beberapa macam teknologi yang berguna untuk membantu pengembangan sistem dalam memenuhi seluruh kebutuhan yang diperlukan. Teknologi-teknologi tersebut yaitu *Codeigniter*, HTML, PHP, MySQL, dan Javascript.

### 2.4.1 *Codeigniter*

*Codeigniter* merupakan salah satu *framework* PHP yang dapat memberikan bantuan kepada pengembang perangkat lunak dalam mempercepat pengembangan aplikasi web berbasis PHP, daripada melakukan penulisan semua kode program dari awal (Hakim, 2010). *Codeigniter* dibangun berbasis MVC (*model, view, controller*). Menurut Sommerville (2011), MVC merupakan sebuah bentuk pemrograman yang memisahkan antara presentasi dan interaksi dari data sistem. Bagian-bagian dari MVC menurut Sommerville (2011), antara lain:

1. Model, yaitu bagian kode program yang menangani data sistem dan operasi terkait dengan data sistem tersebut.
2. View, yaitu bagian kode program yang menetapkan dan mengatur bagaimana data disajikan ke *user*.
3. Controller, yaitu bagian yang mengatur interaksi *user* dan melemparkan interaksi tersebut ke bagian *view* dan *model*.

### 2.4.2 HTML

Menurut Anhar (2010), HTML (*Hyper Text Markup Language*) adalah kumpulan dari beberapa *tag* atau simbol yang dituliskan dalam sebuah file, dimana file tersebut dapat digunakan untuk menampilkan halaman pada web *browser*. Dengan menggunakan beberapa instruksi yang ada di dalam HTML, maka memungkinkan pengembang untuk melakukan fungsi-fungsi seperti membuat

*links* (tautan), menentukan ukuran teks, mengintegrasikan gambar dengan teks, dan membuat form interaktif.

### **2.4.3 PHP**

PHP (*Hypertext Preprocessor*) dapat digunakan sebagai *script server-side* dalam melakukan kegiatan untuk mengembangkan suatu *web*, dimana kode dari PHP disisipkan pada dokumen HTML (Peranginangin, 2006). PHP dikatakan sebagai sebuah *server-side embedded script language* yang berarti sintaks-sintaks dan perintah yang kita berikan akan sepenuhnya dijalankan oleh *server*, namun tetap dilakukan penyertaan pada *web browser*.

### **2.4.4 MySQL**

Menurut Huda (2010), MySQL merupakan suatu bentuk dari perangkat lunak sistem manajemen basis data SQL, yaitu sebutan lainnya ialah DBMS (*Database Management System*). MySQL digunakan untuk membuat dan mengelola suatu database secara terstruktur dan otomatis dengan menggunakan suatu bahasa khusus.

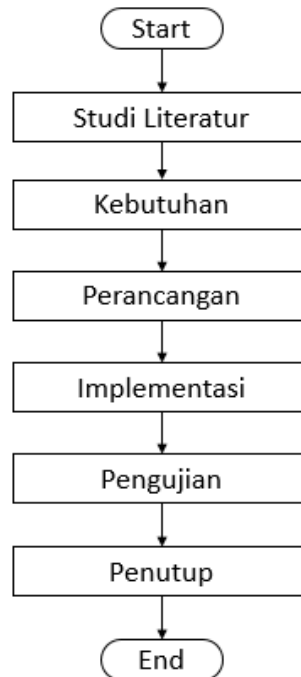
### **2.4.5 Javascript**

Menurut Robbins (2018), javascript merupakan bahasa pemrograman yang menambah interaktivitas dan perilaku kepada suatu situs. Javascript berjalan pada mesin yang digunakan pengguna, bukan pada server yang digunakan. Javascript bergantung pada kemampuan dan pengaturan *browser* yang digunakan pengguna.



## BAB 3 METODOLOGI PENELITIAN

Metodologi penelitian yang dilakukan ini melalui beberapa tahapan-tahapan yang diilustrasikan pada Gambar 3.1.



**Gambar 3.1 Diagram alir metodologi penelitian**

### 3.1 Studi Literatur

Tahap ini digunakan untuk mendapatkan informasi tambahan yang digunakan sebagai acuan, dengan cara mempelajari literatur dari bidang ilmu yang bersangkutan melalui narasumber, jurnal, maupun internet dengan sumber yang sudah terpercaya.

### 3.2 Kebutuhan

Tahap kebutuhan dilakukan untuk mengidentifikasi seluruh kebutuhan pengguna, baik kebutuhan fungsional maupun kebutuhan non-fungsional. Metode pendekatan yang digunakan adalah *Object Oriented Analysis* (OOA). Tahap kebutuhan dimulai dengan melakukan elisitasi dan analisis kebutuhan. Elisitasi dan analisis kebutuhan dilakukan untuk mengetahui permasalahan yang ingin diatasi, mengidentifikasi aktor yang akan terlibat pada sistem, dan mendefinisikan kebutuhan sistem yang dikembangkan. Proses elisitasi kebutuhan dilakukan dengan menggunakan teknik wawancara. Wawancara dilakukan dengan dokter yang bekerja pada klinik Dr. Syarif. Setelah melakukan elisitasi dan analisis kebutuhan, proses berikutnya adalah melakukan spesifikasi kebutuhan. Spesifikasi kebutuhan dilakukan untuk mengetahui apa saja kebutuhan yang terdapat pada sistem dan siapa saja yang dapat melakukan kebutuhan tersebut. Spesifikasi kebutuhan dimodelkan dengan menggunakan *use case diagram*. *Use case* pada

*use case diagram* tersebut kemudian dijelaskan lebih detail dengan menggunakan *use case scenario*.

### 3.3 Perancangan

Perancangan dilakukan berdasarkan hasil dari tahap kebutuhan yang telah dilakukan sebelumnya. Perancangan pada penelitian ini menggunakan metode pendekatan *Object Oriented Design* (OOD). Perancangan ini meliputi:

1. Perancangan arsitektural, yang terdiri dari:
  - a. Pemodelan *sequence diagram*. Tahap ini menjelaskan mengenai urutan interaksi yang dilakukan oleh aktor dan sistem, dan objek-objek di dalam sistem dalam suatu fungsional sistem. Tahap ini juga menampilkan informasi mengenai operasi-operasi yang dilakukan oleh objek-objek tersebut dan data-data yang diperlukan dalam operasi tersebut.
  - b. Pemodelan *class diagram*. Tahap ini menjelaskan mengenai klas-klas yang ada di dalam pengembangan sistem. Klas-klas tersebut memiliki atribut, operasi dan relasi antarklas. Klas-klas tersebut dibuat berdasarkan *sequence diagram* yang telah dibuat sebelumnya. *Class diagram* ini terdiri dari klas *boundary* sebagai klas yang langsung berinteraksi dengan aktor, klas *controller* sebagai klas yang menghubungkan *boundary* dengan *entity*, dan klas *entity* sebagai klas yang melakukan penyimpanan atau mekanisme yang menangkap informasi atau pengetahuan di dalam sistem.
2. Perancangan data. Tahap ini menjelaskan mengenai pendefinisian struktur data yang akan digunakan dalam implementasi sistem. Perancangan data dimodelkan dengan *Conceptual Data Model* (CDM) dan *Physical Data Model* (PDM).
3. Perancangan komponen. Tahap ini menjelaskan mengenai pembuatan algoritme-algoritme dari klas yang telah dirancang pada perancangan arsitektural. Algoritme menggunakan bahasa *pseudocode*. Perancangan komponen ini hanya akan menampilkan beberapa algoritme sebagai sampel. Algoritme yang dijadikan sampel merupakan algoritme dengan prioritas tinggi di dalam sistem.
4. Perancangan antarmuka, yaitu merancang antarmuka sistem yang akan dibangun.

### 3.4 Implementasi

Tahap Implementasi dilakukan terhadap rancangan sistem yang telah dibuat. Tahap implementasi pada penelitian ini menggunakan metode pendekatan *Object Oriented Programming* (OOP). Implementasi sistem berbasis web, menggunakan bahasa pemrograman PHP, menggunakan *framework Codeigniter* dan MySQL sebagai basis data. Hasil yang didapatkan pada tahap ini adalah spesifikasi dari sistem yang digunakan dalam melakukan pengembangan sistem, batasan implementasi dari sistem yang dikembangkan, implementasi data, implementasi

kode program berdasarkan algoritme pada tahap perancangan, dan implementasi antarmuka sistem.

### 3.5 Pengujian

Pengujian dilakukan untuk memeriksa apakah sistem yang telah dibangun sudah berjalan dengan baik, tidak ada kesalahan yang terjadi dan telah sesuai dengan yang diharapkan. Pengujian terdiri dari:

1. Pengujian Unit. Tahap ini menjelaskan mengenai pengujian terhadap unit-unit sistem, untuk mengetahui apakah unit tersebut telah berjalan dengan benar dan layak untuk digunakan atau tidak. Unit yang diuji adalah kelas, dengan cara menguji *method*-nya. Pengujian unit dilakukan dengan menggunakan teknik *white box testing*. Metode pengujian *white box* yang digunakan adalah *basis path testing*. Pengujian *basis path* akan menguji jalur kode program berdasarkan algoritme masing-masing *method* pada setiap kelas. Pengujian unit akan dilakukan pada tiga sampel uji.
2. Pengujian integrasi. Tahap ini menjelaskan mengenai pengujian terhadap gabungan unit yang saling berhubungan dalam kesatuan fungsional sistem. Pengujian ini akan dilakukan dengan menggunakan teknik *white box* dengan *basis path testing* sebagai metodenya. Pengujian *basis path testing* akan menguji jalur kode program berdasarkan algoritme masing-masing *method* pada setiap kelas. Pengujian integrasi menggunakan pendekatan *bottom-up integration*. Pengujian integrasi akan dilakukan pada tiga sampel algoritme.
3. Pengujian validasi. Tahap ini menguji seluruh fungsional sistem, apakah telah mengimplementasikan seluruh kebutuhan yang telah dibuat pada tahap kebutuhan atau tidak. Tahap ini juga menguji apakah keluaran dari suatu fungsional sistem telah sesuai dengan yang didefinisikan pada tahap kebutuhan atau tidak. Pengujian validasi dilakukan dengan menggunakan teknik *black box*. Metode pengujian *black box* yang digunakan adalah *scenario-based testing*.

### 3.6 Penutup

Tahap ini berisi kesimpulan dan saran. Kesimpulan diperoleh berdasarkan hasil pada tahap kebutuhan, perancangan & implementasi, dan pengujian. Kesimpulan tersebut menjawab rumusan masalah yang telah dibuat sebelumnya. Saran merupakan masukan yang berguna untuk pengembangan sistem berikutnya agar dapat berjalan dengan lebih baik lagi.

## BAB 4 KEBUTUHAN

### 4.1 Elisitasi dan Analisis Kebutuhan

Teknik elisitasi kebutuhan yang dilakukan dalam penggalan kebutuhan pada klinik Dr. Syarif adalah tradisional, yaitu wawancara. Topik wawancara yang dilakukan pada klinik Dr. Syarif adalah mengenai prosedur pengelolaan rekam medis pasien. Pihak yang diwawancarai adalah dokter umum yang bekerja pada klinik tersebut, bernama Dr. Syarif Ismail. Analisis kebutuhan dilakukan berdasarkan hasil dari proses elisitasi kebutuhan. Analisis kebutuhan menghasilkan identifikasi aktor dan daftar kebutuhan sistem.

#### 4.1.1 Identifikasi Aktor

Identifikasi aktor bertujuan untuk memberikan gambaran mengenai pengguna yang akan terlibat pada sistem. Penjelasan mengenai siapa saja aktor yang akan menggunakan sistem dapat dilihat pada Tabel 4.1.

**Tabel 4.1 Identifikasi aktor**

Aktor	Deskripsi
Pengguna	Pengguna merupakan aktor yang belum masuk ke dalam sistem.
Admin	Admin berperan dalam mengelola data pengguna sistem.
Petugas Administrasi	Petugas administrasi berperan dalam mengelola rekam medis pasien.
Dokter	Dokter berperan dalam melakukan pemeriksaan terhadap pasien dan menambahkan hasil pemeriksaan ke dalam rekam medis pasien.
Apoteker	Apoteker berperan dalam mengelola data obat.

#### 4.1.2 Daftar Kebutuhan Sistem

Dalam penelitian ini, seluruh kebutuhan fungsional diberi kode SIPREM-F-XX. SIPREM adalah singkatan dari Sistem Pengelolaan Rekam Medis. F merupakan singkatan dari Fungsional dan XX merupakan nomor definisi kebutuhan. Tabel 4.2 menunjukkan daftar kebutuhan fungsional sistem.

**Tabel 4.2 Daftar kebutuhan fungsional**

Kode	Kebutuhan Fungsional	Use Case	Aktor
SIPREM-F-01	Sistem harus mampu menyediakan fungsi <i>login</i> untuk mengizinkan pengguna mengakses sistem dengan cara memasukkan <i>username</i> dan <i>password</i> .	Login	Pengguna
SIPREM-F-02	Sistem harus mampu menyediakan fungsi <i>logout</i> agar pengguna yang telah terhubung dengan sistem dapat keluar dari sistem.	<i>Logout</i>	Petugas Administrasi, Dokter, Apoteker, Admin
SIPREM-F-03	Sistem harus mampu menyediakan fungsi menambah pengguna dengan memasukkan nama, jenis pengguna, <i>username</i> , dan <i>password</i> .	Menambah Pengguna	Admin
SIPREM-F-04	Sistem harus mampu menyediakan fungsi menampilkan daftar pengguna sistem yang terdiri dari nomor, nama pengguna, jenis pengguna, <i>username</i> , dan <i>password</i> .	Melihat Daftar Pengguna	Admin
SIPREM-F-05	Sistem harus mampu menyediakan fungsi mengubah data pengguna yang terdiri dari nama pengguna, jenis pengguna, <i>username</i> , dan <i>password</i> .	Mengubah Data Pengguna	Admin
SIPREM-F-06	Sistem harus mampu menyediakan fungsi menampilkan daftar pasien khusus Petugas Administrasi yang terdiri dari nomor, nama pasien,	Melihat Daftar Pasien Khusus Petugas Administrasi	Petugas Administrasi

	nomor BPJS, jenis kelamin, tempat lahir, tanggal lahir, alamat, dan nomor telepon serta kolom pencarian.		
SIPREM-F-07	Sistem harus mampu menyediakan fungsi menampilkan daftar pasien khusus dokter yang terdiri dari nomor, nama pasien, nomor BPJS, jenis kelamin, tempat lahir, tanggal lahir, alamat, dan nomor telepon serta kolom pencarian.	Melihat Daftar Pasien Khusus Dokter	Dokter
SIPREM-F-08	Sistem harus mampu menyediakan fungsi menambah pasien yang terdiri dari nama pasien, nomor BPJS, jenis kelamin, tempat lahir, tanggal lahir, alamat, dan nomor telepon.	Menambah Pasien Baru	Petugas Administrasi
SIPREM-F-09	Sistem harus mampu menyediakan fungsi mengubah data pasien yang terdiri dari nomor, nama pasien, nomor BPJS, jenis kelamin, tempat lahir, tanggal lahir, alamat, dan nomor telepon .	Mengubah data pasien	Petugas Administrasi
SIPREM-F-10	Sistem harus mampu menyediakan fungsi menampilkan rekam medis pasien yang terdiri dari nama pasien, jenis kelamin, nomor BPJS, alamat, tempat lahir, tanggal lahir, nomor telepon, nomor, dokter pemeriksa, tanggal pemeriksaan, waktu pemeriksaan, anamnesa, pemeriksaan fisik,	Melihat Rekam Medis	Dokter

	diagnosis, terapi, dan catatan.		
SIPREM-F-11	Sistem harus mampu menyediakan fungsi menampilkan daftar obat yang terdiri dari nomor, nama obat, deskripsi obat, harga obat, satuan, dan kuantitas obat serta kolom pencarian .	Melihat daftar obat khusus dokter	Apoteker
SIPREM-F-12	Sistem harus mampu menyediakan fungsi menampilkan daftar obat yang terdiri dari nomor, nama obat, deskripsi obat, harga obat, satuan, dan kuantitas obat serta kolom pencarian.	Melihat Daftar Obat Khusus Apoteker	Apoteker
SIPREM-F-13	Sistem harus mampu menyediakan fungsi menambah obat baru yang terdiri dari nama obat, deskripsi obat, harga obat, satuan, dan kuantitas obat.	Menambah obat baru	Apoteker
SIPREM-F-14	Sistem harus mampu menyediakan fungsi mengubah data obat yang terdiri dari nama obat, deskripsi obat, harga obat, satuan, dan kuantitas obat.	Mengubah data obat	Apoteker
SIPREM-F-15	Sistem harus mampu menyediakan fungsi menyimpan hasil pemeriksaan pasien yang terdiri dari anamnesa, pemeriksaan fisik, diagnosis, terapi, dan catatan.	Menambahka n hasil pemeriksaan	Dokter
SIPREM-F-16	Sistem harus mampu menyediakan fungsi mencetak rekam medis yang terdiri dari nama pasien, jenis kelamin,	Mencetak rekam medis	dokter

	nomor BPJS, alamat, tempat lahir, tanggal lahir, nomor telepon, nomor, dokter pemeriksa, tanggal pemeriksaan, waktu pemeriksaan, anamnesa, pemeriksaan fisik, diagnosis, terapi, dan catatan.		
SIPREM-F-17	Sistem harus mampu menyediakan fungsi menampilkan antrian pasien yang terdiri dari nomor antrian, nama pasien, nomor bpjs, tanggal mendaftar, jam mendaftar, jam pemeriksaan, dan status pasien.	Melihat antrian khusus petugas administrasi	Petugas Administrasi
SIPREM-F-18	Sistem harus mampu menyediakan fungsi menambahkan pasien ke dalam antrian.	Menambahkan Pasien Ke dalam Antrian	Petugas Administrasi
SIPREM-F-19	Sistem harus mampu menyediakan fungsi membatalkan antrian pasien.	Membatalkan Antrian Pasien	Petugas Administrasi
SIPREM-F-20	Sistem harus mampu menyediakan fungsi menambahkan pasien ke dalam ruang pemeriksaan.	Menambahkan pasien ke dalam ruang pemeriksaan	Petugas Administrasi

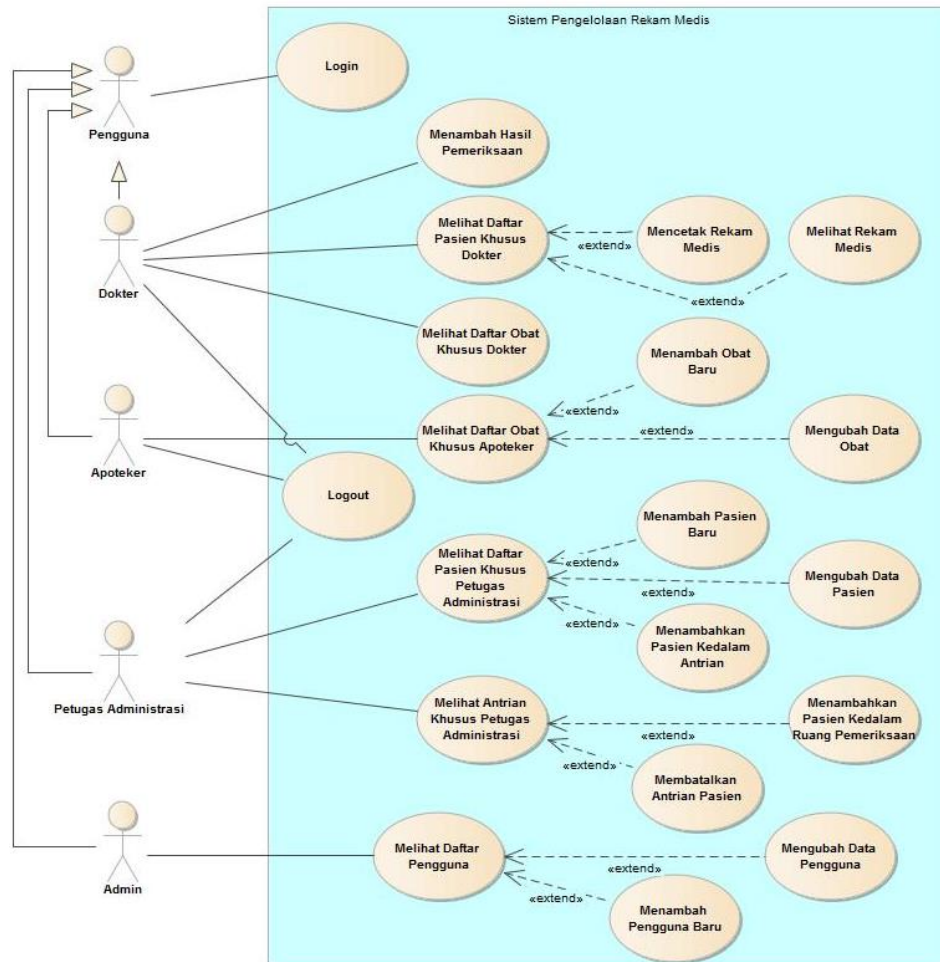
## 4.2 Spesifikasi Kebutuhan

Spesifikasi kebutuhan mendefinisikan apa saja kebutuhan yang terdapat pada sistem dan siapa saja aktor yang dapat melakukan kebutuhan tersebut. Spesifikasi kebutuhan dimodelkan dengan menggunakan *use case diagram* dan *use case scenario*.



### 4.2.1 Use Case Diagram

*Usecase Diagram* merupakan diagram yang berguna dalam memberikan gambaran mengenai aktor-aktor yang terdapat pada sistem dan apa saja yang dapat dilakukan aktor tersebut. *Use case diagram* sistem diilustrasikan pada Gambar 4.1.



**Gambar 4.1 Use case diagram Sistem Pengelolaan Rekam Medis Pasien Berbasis Web**

### 4.2.2 Use Case Scenario

Setiap *use case* yang dimodelkan akan dijelaskan secara detail pada *use case scenario*.

#### 4.2.2.1 Use Case Scenario Login

Tabel 4.3 merupakan *use case scenario* untuk kebutuhan *login*.

**Tabel 4.3 Use case scenario Login**

SIPREM-F-01	Login
Aktor	Pengguna
Objective	Memungkinkan pengguna untuk masuk ke dalam sistem.
Pre-Condition	Aktor mengunjungi halaman login
Main Flow	<ol style="list-style-type: none"><li>1. Aktor memasukkan <i>username</i> &amp; <i>password</i> lalu menekan tombol "<i>login</i>"</li><li>2. Sistem melakukan otentikasi <i>username</i> dan <i>password</i> pada <i>database</i></li></ol>
Alternative Flows	<ol style="list-style-type: none"><li>1.a Jika pengguna menekan tombol "<i>login</i>" tanpa mengisi kolom <i>username</i> atau <i>password</i>, maka sistem akan menampilkan pesan untuk segera mengisi kolom yang kosong</li><li>2.a Jika <i>username</i> salah, maka sistem akan menampilkan pesan bahwa <i>username</i> yang dimasukkan salah</li><li>2.b Jika <i>password</i> salah, maka sistem akan menampilkan pesan bahwa <i>password</i> yang dimasukkan salah</li></ol>
Post-Condition	Aktor berhasil teridentifikasi sebagai Admin atau Dokter atau Petugas Administrasi atau Apoteker dan halaman utama ditampilkan

#### 4.2.2.2 Use Case Scenario Logout

Tabel 4.4 merupakan *use case scenario* untuk kebutuhan *Logout*.

**Tabel 4.4 Use case scenario Logout**

SIPREM-F-02	Logout
Aktor	Petugas Administrasi, Dokter, Apoteker, Admin
Objective	Memungkinkan pengguna untuk keluar dari sistem

<i>Pre-Condition</i>	Aktor sudah teridentifikasi sebagai Petugas Administrasi, dokter, apoteker, atau admin
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol <i>logout</i> yang terdapat pada <i>Navigation Bar</i></li> <li>2. Sistem menampilkan pilihan konfirmasi “yes” dan “cancel”</li> <li>3. Aktor menekan tombol “yes”</li> </ol>
<i>Alternative Flows</i>	3.a Aktor menekan tombol “cancel”, maka aktor tidak keluar dari sistem.
<i>Post-Condition</i>	Aktor berhasil <i>logout</i> dari sistem dan menampilkan halaman <i>login</i>

#### 4.2.2.3 Use Case Scenario Menambah Pengguna

Tabel 4.5 merupakan *use case scenario* untuk kebutuhan menambah pengguna.

**Tabel 4.5 Use case scenario Menambah Pengguna**

SIPREM-F-03	Menambah Pengguna
Aktor	Admin
<i>Objective</i>	Memungkinkan aktor untuk menambah pengguna sistem
<i>Pre-Condition</i>	Aktor teridentifikasi sebagai Admin dan halaman daftar pengguna telah ditampilkan
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol “tambah pengguna”</li> <li>2. Sistem menampilkan form untuk menambah pengguna yang terdiri dari nama pengguna, username, <i>password</i>, dan pekerjaan serta tombol <i>submit</i>.</li> <li>3. Aktor mengisi form secara lengkap lalu menekan tombol <i>submit</i></li> </ol>
<i>Alternative Flows</i>	<p>3.a Jika terdapat kolom kosong pada form dan aktor menekan tombol <i>submit</i>, maka muncul pesan untuk harap mengisi kolom yang kosong</p> <p>3.b Jika <i>username</i> telah digunakan, maka muncul pesan bahwa <i>username</i> telah digunakan dan kembali ke halaman tambah pengguna</p>

<i>Post-Condition</i>	Data pengguna baru berhasil disimpan pada sistem dan menampilkan halaman daftar pengguna
-----------------------	--

#### 4.2.2.4 Use Case Scenario Melihat Daftar Pengguna

Tabel 4.6 merupakan *use case scenario* untuk kebutuhan melihat daftar pengguna.

**Tabel 4.6 Use case scenario Melihat Daftar Pengguna**

SIPREM-F-04	Melihat Daftar Pengguna
Aktor	Admin
<i>Objective</i>	Memungkinkan aktor untuk melihat daftar pengguna sistem
<i>Pre-Condition</i>	Aktor teridentifikasi sebagai Admin
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu “Lihat Pengguna” yang terletak pada <i>menu bar</i></li> <li>2. Sistem menampilkan daftar pengguna sistem yang terdiri dari nomor, nama pengguna, pekerjaan, status, <i>username</i>, dan <i>password</i> dalam bentuk tabel serta tombol ubah data pengguna pada tiap baris data pengguna dan tombol tambah pengguna di atas tabel daftar pengguna</li> </ol>
<i>Alternative Flows</i>	<ol style="list-style-type: none"> <li>2.a Perluasan dari <i>use case</i> Menambah Pengguna dengan kode kebutuhan SIPREM-F-03</li> <li>2.b Perluasan dari <i>use case</i> Mengubah Data Pengguna dengan kode kebutuhan SIPREM-F-05</li> </ol>
<i>Post-Condition</i>	Halaman daftar pengguna sistem berhasil ditampilkan.

#### 4.2.2.5 Use Case Scenario Mengubah Data Pengguna

Tabel 4.7 merupakan *use case scenario* untuk kebutuhan mengubah data obat.

**Tabel 4.7 Use case scenario Mengubah Data Pengguna**

SIPREM-F-05	Mengubah Data Pengguna
-------------	------------------------

Aktor	Admin
<i>Objective</i>	Memungkinkan aktor untuk mengubah data pengguna yang terdapat pada sistem
<i>Pre-Condition</i>	Aktor teridentifikasi sebagai Admin dan halaman daftar pengguna telah ditampilkan
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol ubah data pengguna pada pengguna yang datanya ingin dirubah</li> <li>2. Sistem menampilkan form untuk mengubah data pengguna yang terdiri dari nama pengguna, <i>username</i>, <i>password</i>, pekerjaan, dan status serta tombol <i>submit</i></li> <li>3. Aktor mengubah nama pengguna, <i>username</i>, <i>password</i>, pekerjaan, atau status dan menekan tombol <i>submit</i></li> <li>4. Sistem menyimpan perubahan data pengguna ke dalam sistem</li> </ol>
<i>Alternative Flows</i>	<ol style="list-style-type: none"> <li>3.a Jika aktor menekan tombol <i>cancel</i>, maka sistem menampilkan halaman utama</li> <li>4.a Jika terdapat kolom kosong pada form dan aktor menekan tombol <i>submit</i>, maka muncul pesan untuk harap mengisi kolom yang kosong</li> <li>4.b jika data yang dirubah adalah <i>username</i> dan <i>username</i> tersebut telah digunakan oleh pengguna lain, maka sistem menampilkan pesan bahwa <i>username</i> telah digunakan dan kembali ke halaman ubah data pengguna</li> </ol>
<i>Post-Condition</i>	data pengguna sistem yang dipilih berhasil dirubah dan halaman daftar pengguna ditampilkan

#### 4.2.2.6 Use Case Scenario Melihat Daftar Pasien Khusus Petugas Administrasi

Tabel 4.8 merupakan *use case scenario* untuk kebutuhan melihat daftar pasien khusus petugas administrasi.

**Tabel 4.8 Use case scenario Melihat Daftar Pasien Khusus Petugas**

##### **Administrasi**

SIPREM-F-06	Melihat Daftar Pasien Khusus Petugas Administrasi
Aktor	Petugas Administrasi

<i>Objective</i>	Memungkinkan aktor untuk melihat daftar pasien
<i>Pre-Condition</i>	Aktor teridentifikasi sebagai Petugas Administrasi
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu lihat pasien yang terdapat pada <i>menu bar</i></li> <li>2. Sistem menampilkan daftar pasien yang terdiri dari nomor, nama pasien, nomor BPJS, Jenis Kelamin, tempat lahir, tanggal lahir, alamat, nomor telepon dan kolom pencarian serta tombol ubah data pasien dan tombol masukkan pasien ke dalam antrian pada tiap baris data pasien dan tombol tambah pasien di atas tabel daftar pasien.</li> </ol>
<i>Alternative Flows</i>	<ol style="list-style-type: none"> <li>2.a Perluasan dari <i>use case</i> Menambah Pasien Baru dengan kode kebutuhan SIPREM-F-08</li> <li>2.b Perluasan dari <i>use case</i> Mengubah Data Pasien dengan kode kebutuhan SIPREM-F-09</li> <li>2.c Perluasan dari <i>use case</i> Menambahkan Pasien Ke dalam Antrian dengan kode kebutuhan SIPREM-F-18</li> </ol>
<i>Post-Condition</i>	Halaman daftar pasien berhasil ditampilkan

#### 4.2.2.7 Use Case Scenario Melihat Daftar Pasien Khusus Dokter

Tabel 4.9 merupakan *use case scenario* untuk kebutuhan melihat daftar pasien khusus dokter.

**Tabel 4.9 Use case scenario Melihat Daftar Pasien Khusus Dokter**

SIPREM-F-07	Melihat Daftar Pasien Khusus Dokter
Aktor	Dokter
<i>Objective</i>	Memungkinkan aktor untuk melihat daftar pasien
<i>Pre-Condition</i>	Aktor teridentifikasi sebagai dokter
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu lihat pasien yang terdapat pada <i>menu bar</i></li> <li>2. Sistem menampilkan daftar pasien yang terdiri dari nomor, nama pasien, nomor BPJS, Jenis Kelamin, tempat lahir, tanggal lahir, alamat, nomor telepon dan kolom pencarian serta tombol lihat rekam medis dan tombol cetak rekam medis di tiap baris data pasien.</li> </ol>

<i>Alternative Flows</i>	2.a Perluasan dari <i>use case</i> Melihat Rekam Medis dengan kode kebutuhan SIPREM-F-10 2.b Perluasan dari <i>use case</i> Mencetak Rekam Medis dengan kode kebutuhan SIPREM-F-16
<i>Post-Condition</i>	Halaman daftar pasien berhasil ditampilkan

#### 4.2.2.8 Use Case Scenario Menambah Pasien Baru

Tabel 4.10 merupakan *use case scenario* untuk kebutuhan menambah pasien baru.

**Tabel 4.10 Use case scenario Menambah Pasien Baru**

SIPREM-F-8	Menambah Pasien Baru
Aktor	Petugas Administrasi
<i>Objective</i>	Memungkinkan aktor untuk menambah pasien baru ke dalam sistem
<i>Pre-Condition</i>	Aktor teridentifikasi sebagai Petugas Administrasi dan halaman daftar pasien telah ditampilkan
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol tambah pasien</li> <li>2. Sistem menampilkan form yang terdiri dari nama pasien, nomor BPJS, jenis kelamin, tempat lahir, tanggal lahir, alamat, dan nomor telepon serta tombol <i>submit</i> dan <i>cancel</i>.</li> <li>3. Aktor mengisi form secara lengkap dan menekan tombol <i>submit</i></li> </ol>
<i>Alternative Flows</i>	<ol style="list-style-type: none"> <li>3.a Jika terdapat kolom kosong pada form dan aktor menekan tombol <i>submit</i>, maka muncul pesan untuk harap mengisi kolom yang kosong dan sistem tidak jadi menyimpan data pasien baru</li> <li>3.b Jika nomor BPJS pasien memiliki kesamaan dengan nomor BPJS pasien lain, maka sistem menampilkan informasi bahwa nomor BPJS sudah digunakan</li> <li>3.c Jika aktor menekan tombol <i>cancel</i>, maka sistem menampilkan halaman utama</li> </ol>
<i>Post-Condition</i>	Pasien baru berhasil disimpan oleh sistem dan menampilkan halaman daftar pasien

#### 4.2.2.9 Use Case Scenario Mengubah Data Pasien

Tabel 4.11 merupakan *use case scenario* untuk kebutuhan mengubah data pasien.

**Tabel 4.11 Use case scenario Mengubah Data Pasien**

SIPREM-F-09	Mengubah Data Pasien
Aktor	Petugas Administrasi
<i>Objective</i>	Memungkinkan aktor untuk mengubah data pasien yang terdapat pada sistem
<i>Pre-Condition</i>	Aktor teridentifikasi sebagai Petugas Administrasi dan halaman daftar pasien telah ditampilkan
<i>Main Flow</i>	<ol style="list-style-type: none"><li>1. Aktor menekan tombol ubah data pasien pada pasien yang datanya ingin diubah</li><li>2. Sistem menampilkan form untuk mengubah data pasien yang terdiri dari nama pasien, nomor BPJS, Jenis Kelamin, tempat lahir, tanggal lahir, alamat, dan nomor telepon serta tombol <i>submit</i> dan <i>cancel</i>.</li><li>3. Aktor mengubah data pasien dan menekan tombol <i>submit</i>.</li></ol>
<i>Alternative Flows</i>	<ol style="list-style-type: none"><li>3.a Jika terdapat kolom kosong pada form dan aktor menekan tombol <i>submit</i>, maka muncul pesan untuk harap mengisi kolom yang kosong</li><li>3.b Jika nomor BPJS pasien dirubah dan memiliki kesamaan dengan nomor BPJS pasien lain, maka sistem menampilkan informasi bahwa nomor BPJS telah digunakan</li><li>3.c Jika aktor menekan tombol <i>cancel</i>, maka sistem menampilkan halaman utama</li></ol>
<i>Post-Condition</i>	Data pasien berhasil disimpan oleh sistem dan halaman daftar pasien ditampilkan



#### 4.2.2.10 Use Case Scenario Melihat Rekam Medis

Tabel 4.12 merupakan *use case scenario* untuk kebutuhan melihat rekam medis.

**Tabel 4.12 Use case scenario Melihat Rekam Medis**

SIPREM-F-10	Melihat Rekam Medis
Aktor	Dokter
<i>Objective</i>	Memungkinkan aktor untuk melihat rekam medis pasien
<i>Pre-Condition</i>	Aktor teridentifikasi sebagai dokter dan halaman daftar pasien telah ditampilkan
<i>Main Flow</i>	<ol style="list-style-type: none"><li>1. aktor menekan tombol lihat rekam medis</li><li>2. sistem menampilkan data rekam medis pasien yang terdiri dari nama pasien, nomor BPJS, jenis kelamin, tempat lahir, tanggal lahir, dan nomor telepon, serta sebuah tabel yang berisi nomor, dokter pemeriksa, tanggal pemeriksaan, waktu pemeriksaan, anamnesa, pemeriksaan fisik, diagnosis, terapi, dan catatan.</li></ol>
<i>Alternative Flows</i>	2.a Jika belum ada data medis pada rekam medis pasien maka sistem menampilkan tabel tanpa data medis pasien.
<i>Post-Condition</i>	Rekam medis pasien berhasil ditampilkan oleh sistem

#### 4.2.2.11 Use Case Scenario Melihat Daftar Obat Khusus Dokter

Tabel 4.13 merupakan *use case scenario* untuk kebutuhan melihat daftar obat khusus dokter.

**Tabel 4.13 Use case scenario Melihat Daftar Obat Khusus Dokter**

SIPREM-F-11	Melihat Daftar Obat Khusus Dokter
Aktor	Dokter

<i>Objective</i>	Memungkinkan aktor untuk melihat daftar obat yang ada pada sistem
<i>Pre-Condition</i>	Aktor teridentifikasi sebagai dokter
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu lihat obat yang terdapat pada <i>menu bar</i></li> <li>2. Sistem menampilkan daftar obat yang terdiri dari nomor, nama obat, deskripsi obat, harga obat, satuan, dan kuantitas obat dalam bentuk tabel serta kolom pencarian</li> </ol>
<i>Alternative Flows</i>	2.a Jika data obat kosong maka sistem menampilkan tabel dengan data kosong
<i>Post-Condition</i>	Halaman daftar obat berhasil ditampilkan oleh sistem

#### 4.2.2.12 Use Case Scenario Melihat Daftar Obat Khusus Apoteker

Tabel 4.14 merupakan *use case scenario* untuk kebutuhan melihat daftar obat khusus apoteker.

**Tabel 4.14 Use case scenario Melihat Daftar Obat Khusus Apoteker**

<b>SIPREM-F-12</b>	<b>Melihat Daftar Obat Khusus Apoteker</b>
Aktor	Apoteker
<i>Objective</i>	Memungkinkan aktor untuk melihat daftar obat yang ada pada sistem
<i>Pre-Condition</i>	Aktor teridentifikasi sebagai apoteker
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu lihat obat yang terdapat pada <i>menu bar</i></li> <li>2. Sistem menampilkan daftar obat yang terdiri dari nomor, nama obat, deskripsi obat, harga obat, satuan, kuantitas obat dalam bentuk tabel dan kolom pencarian serta tombol ubah data obat yang terdapat pada setiap baris data obat dan tombol tambah obat yang terdapat di atas tabel.</li> </ol>
<i>Alternative Flows</i>	2.a Perluasan dari <i>use case</i> Menambah Obat Baru dengan kode kebutuhan SIPREM-F-13

	2.b Perluasan dari <i>use case</i> Mengubah Data Obat dengan kode kebutuhan SIPREM-F-14
<i>Post-Condition</i>	Daftar obat berhasil ditampilkan oleh sistem

#### 4.2.2.13 Use Case Scenario Menambah Obat Baru

Tabel 4.15 merupakan *use case scenario* untuk kebutuhan menambah obat baru.

**Tabel 4.15 Use case scenario Menambah Obat Baru**

SIPREM-F-13	Menambah Obat Baru
Aktor	Apoteker
<i>Objective</i>	Memungkinkan aktor untuk menambah obat baru ke dalam sistem
<i>Pre-Condition</i>	Aktor teridentifikasi sebagai apoteker dan halaman daftar obat telah ditampilkan
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol tambah obat</li> <li>2. Sistem menampilkan form untuk menambah obat baru yang terdiri dari nama obat, deskripsi obat, harga obat, satuan, dan kuantitas obat serta tombol <i>submit</i> dan tombol <i>cancel</i></li> <li>3. Aktor mengisi form, lalu menekan tombol <i>submit</i></li> </ol>
<i>Alternative Flows</i>	<ol style="list-style-type: none"> <li>3.a Jika terdapat kolom kosong pada form dan Apoteker menekan tombol <i>submit</i>, maka muncul pesan untuk harap mengisi kolom yang kosong</li> <li>3.b Jika nama obat baru memiliki kesamaan dengan nama obat pada <i>database</i>, maka sistem akan menampilkan pesan bahwa nama obat telah digunakan</li> <li>3.c Jika aktor menekan tombol <i>cancel</i>, maka sistem menampilkan halaman utama</li> </ol>
<i>Post-Condition</i>	Obat baru berhasil disimpan oleh sistem dan menampilkan halaman daftar obat

#### 4.2.2.14 Use Case Scenario Mengubah Data Obat

Tabel 4.16 merupakan *use case scenario* untuk kebutuhan mengubah data obat.

**Tabel 4.16 Use case scenario Mengubah Data Obat**

SIPREM-F-14	Mengubah Data Obat
Aktor	Apoteker
<i>Objective</i>	Memungkinkan aktor untuk mengubah data obat yang terdapat pada sistem
<i>Pre-Condition</i>	Aktor teridentifikasi sebagai apoteker dan halaman daftar obat telah ditampilkan
<i>Main Flow</i>	<ol style="list-style-type: none"><li>1. Aktor menekan tombol ubah data obat pada obat yang ingin dirubah datanya</li><li>2. Sistem menampilkan form untuk mengubah data obat yang terdiri dari nama obat, deskripsi obat, harga obat, satuan, dan kuantitas obat serta tombol <i>submit</i> dan tombol <i>cancel</i></li><li>3. Aktor mengubah data obat dan menekan tombol <i>submit</i></li></ol>
<i>Alternative Flows</i>	<ol style="list-style-type: none"><li>3.a jika nama obat dirubah dan memiliki kesamaan dengan nama obat pada <i>database</i>, maka sistem akan menampilkan pesan bahwa nama obat telah digunakan.</li><li>3.b Jika aktor menekan tombol <i>cancel</i>, maka sistem menampilkan halaman daftar obat</li></ol>
<i>Post-Condition</i>	Data obat berhasil disimpan oleh sistem dan halaman daftar obat ditampilkan

#### 4.2.2.15 Use Case Scenario Menambah Hasil Pemeriksaan

Tabel 4.17 merupakan *use case scenario* untuk kebutuhan menambah hasil pemeriksaan.

**Tabel 4.17 Use case scenario Menambah Hasil Pemeriksaan**

SIPREM-F-15	Menambah Hasil Pemeriksaan
Aktor	Dokter

<i>Objective</i>	Memungkinkan aktor untuk menambah hasil pemeriksaan pasien ke dalam rekam medis
<i>Pre-Condition</i>	Aktor teridentifikasi sebagai dokter
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu lihat antrian</li> <li>2. Sistem menampilkan data diri pasien yang terdiri dari nama pasien, jenis kelamin, nomor BPJS, nomor antrian, alamat, tempat lahir, tanggal lahir, dan nomor telepon. Sistem juga menyediakan form untuk menambah hasil pemeriksaan pasien yang terdiri dari anamnesa, pemeriksaan fisik, diagnosis, terapi, dan catatan resep obat serta tombol <i>cancel</i> dan <i>submit</i></li> <li>3. Aktor mengisi form lalu menekan tombol <i>submit</i></li> </ol>
<i>Alternative Flows</i>	3.a Jika Dokter menekan tombol <i>cancel</i> , maka proses penambahan hasil pemeriksaan dibatalkan.
<i>Post-Condition</i>	Hasil pemeriksaan pasien berhasil disimpan oleh sistem dan halaman utama ditampilkan

#### 4.2.2.16 Use Case Scenario Mencetak Rekam Medis

Tabel 4.18 merupakan *use case scenario* untuk kebutuhan mencetak rekam medis.

**Tabel 4.18 Use case scenario Mencetak Rekam Medis**

<b>SIPREM-F-16</b>	<b>Mencetak Rekam Medis</b>
Aktor	Dokter
<i>Objective</i>	Memungkinkan aktor untuk mencetak rekam medis pasien
<i>Pre-Condition</i>	Aktor teridentifikasi sebagai dokter dan halaman daftar pasien telah ditampilkan
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. aktor menekan tombol cetak rekam medis pada pasien yang ingin dicetak rekam medisnya</li> </ol>

	2. sistem menampilkan <i>tab</i> baru yang berisi data diri dan rekam medis pasien 3. aktor menekan tombol <i>print</i>
<i>Alternative Flows</i>	3.a aktor menekan tombol <i>save as pdf</i>
<i>Post-Condition</i>	Rekam medis berhasil dicetak oleh sistem

#### 4.2.2.17 Use Case Scenario Melihat Antrian Pasien Khusus Petugas Administrasi

Tabel 4.19 merupakan *use case scenario* untuk kebutuhan melihat antrian pasien khusus petugas administrasi.

**Tabel 4.19. Use case scenario Melihat Antrian Pasien Khusus Petugas Administrasi**

<b>SIPREM-F-17</b>	<b>Melihat Antrian Pasien Khusus Petugas Administrasi</b>
Aktor	Petugas Administrasi
<i>Objective</i>	Memungkinkan aktor untuk melihat antrian pasien khusus petugas administrasi
<i>Pre-Condition</i>	Aktor teridentifikasi sebagai petugas administrasi
<i>Main Flow</i>	1. Aktor memilih menu lihat antrian yang terdapat pada <i>menu bar</i> 2. Sistem menampilkan daftar antrian pasien yang terdiri dari nomor antrian, nama pasien, nomor BPJS, Tanggal mendaftar, jam mendaftar, jam pemeriksaan, dan status pasien serta tombol batalkan antrian dan tombol masukkan ke dalam ruangan pada baris pasien yang sudah mengantri.
<i>Alternative Flows</i>	2.a Perluasan dari <i>use case</i> Membatalkan Antrian Pasien dengan kode kebutuhan SIPREM-F-19 2.b Perluasan dari <i>use case</i> Menambahkan Pasien Ke dalam Ruang Pemeriksaan dengan kode kebutuhan SIPREM-F-20
<i>Post-Condition</i>	Antrian Pasien Berhasil ditampilkan oleh sistem

#### 4.2.2.18 Use Case Scenario Menambahkan Pasien Ke dalam Antrian

Tabel 4.20 merupakan *use case scenario* untuk kebutuhan menambah pasien ke dalam antrian.

**Tabel 4.20. Use case scenario Menambahkan Pasien ke dalam Antrian**

SIPREM-F-18	Menambahkan Pasien Ke dalam Antrian
Aktor	Petugas Administrasi
<i>Objective</i>	Memungkinkan aktor untuk menambahkan pasien ke dalam antrian
<i>Pre-Condition</i>	Aktor teridentifikasi sebagai petugas administrasi dan halaman daftar pasien telah ditampilkan
<i>Main Flow</i>	<ol style="list-style-type: none"><li>1. Aktor menekan tombol “masukkan ke dalam antrian”</li><li>2. Sistem menampilkan pesan bahwa pasien berhasil dimasukkan ke dalam antrian</li></ol>
<i>Alternative Flows</i>	<ol style="list-style-type: none"><li>2.a. Jika pasien sudah terdaftar dan nomor antrian pasien belum terlewati, maka pasien gagal ditambahkan ke dalam antrian dan sistem menampilkan pesan bahwa pasien sudah terdaftar dan nomor antrian belum terlewati.</li></ol>
<i>Post-Condition</i>	Pasien berhasil ditambahkan ke dalam antrian oleh sistem dan halaman antrian ditampilkan

#### 4.2.2.19 Use Case Scenario Membatalkan Antrian Pasien

Tabel 4.21 merupakan *use case scenario* untuk kebutuhan membatalkan antrian pasien.

**Tabel 4.21. Use case scenario Membatalkan Antrian Pasien**

SIPREM-F-19	Membatalkan Antrian Pasien
Aktor	Petugas Administrasi
<i>Objective</i>	Memungkinkan aktor untuk membatalkan antrian pasien

<i>Pre-Condition</i>	Aktor teridentifikasi sebagai petugas administrasi dan halaman daftar antrian pasien telah ditampilkan
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol batalkan antrian</li> <li>2. Sistem menampilkan <i>pop-up</i> yang berisi pilihan ya atau tidak</li> <li>3. Aktor menekan tombol ya</li> </ol>
<i>Alternative Flows</i>	3a. Jika aktor menekan tombol tidak, maka pasien tetap berada pada antrian.
<i>Post-Condition</i>	Antrian pasien berhasil dibatalkan oleh sistem dan halaman daftar antrian pasien ditampilkan

#### 4.2.2.20 Use Case Scenario Menambahkan Pasien Ke dalam Ruang Pemeriksaan

Tabel 4.22 merupakan *use case scenario* untuk kebutuhan menambahkan pasien ke dalam ruang pemeriksaan.

**Tabel 4.22. Use case scenario Menambahkan Pasien Ke dalam Ruang Pemeriksaan**

SIPREM-F-20	Menambahkan Pasien Ke dalam Ruang Pemeriksaan
Aktor	Petugas Administrasi
<i>Objective</i>	Memungkinkan aktor untuk menambahkan pasien ke dalam ruang pemeriksaan
<i>Pre-Condition</i>	Aktor teridentifikasi sebagai petugas administrasi dan halaman daftar antrian pasien telah ditampilkan
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol masukkan ke dalam ruangan</li> <li>2. Sistem menampilkan <i>pop-up</i> yang berisi pilihan untuk masukkan pasien atau <i>cancel</i></li> <li>3. Aktor menekan tombol masukkan pasien</li> </ol>
<i>Alternative Flows</i>	3a. Jika aktor menekan tombol <i>cancel</i> , maka pasien tidak jadi dimasukkan ke dalam ruang pemeriksaan



	<p>3.b Jika di dalam ruangan pemeriksaan masih terdapat pasien yang lain, maka tombol untuk memasukkan pasien ke dalam ruang pemeriksaan tidak bisa ditekan.</p> <p>3.c Jika dokter belum masuk ke dalam sistem, maka tombol untuk memasukkan pasien ke dalam ruang pemeriksaan tidak bisa ditekan.</p>
<i>Post-Condition</i>	Pasien berhasil ditambahkan ke dalam ruang pemeriksaan oleh sistem dan halaman daftar antrian pasien ditampilkan

## BAB 5 PERANCANGAN DAN IMPLEMENTASI

### 5.1 Perancangan

Setelah melakukan tahap kebutuhan, tahap berikutnya adalah perancangan sistem. Perancangan sistem dilakukan berdasarkan hasil dari tahap kebutuhan. Proses perancangan sistem dibagi menjadi beberapa tahap, yaitu perancangan arsitektural, perancangan data, perancangan komponen, dan perancangan antarmuka.

#### 5.1.1 Perancangan Arsitektural

Perancangan arsitektural menjelaskan mengenai *sequence diagram* dan *class diagram*. *Sequence diagram* menggambarkan interaksi antarobjek pada sistem dalam suatu fungsional sistem. *Sequence diagram* yang akan dijelaskan adalah *sequence diagram* menambah pengguna, menambah pasien, dan menambah obat. *Class diagram* akan memodelkan klas-klas yang akan membentuk sistem beserta atribut dan operasi serta relasi antarklas.

##### 5.1.1.1 Pemodelan *Sequence Diagram*

Pemodelan *sequence diagram* menjelaskan *sequence diagram* dari tiga fitur utama sistem, yaitu menambah pengguna, menambah pasien baru, dan menambah obat baru.

##### ***Sequence Diagram* Menambah Pengguna**

Pada *sequence diagram* menambah pengguna, aktor yang berperan adalah admin. Dua *boundary* yang terlibat adalah `daftarPengguna.php` dan `tambahPengguna.php`. Objek *controller* yang terlibat adalah `PenggunaController`. Dua objek *entity* yang terlibat adalah `Pengguna` dan `Pekerjaan`. *Sequence diagram* menambah pengguna diilustrasikan pada Gambar 5.1.

##### ***Sequence Diagram* Menambah Pasien Baru**

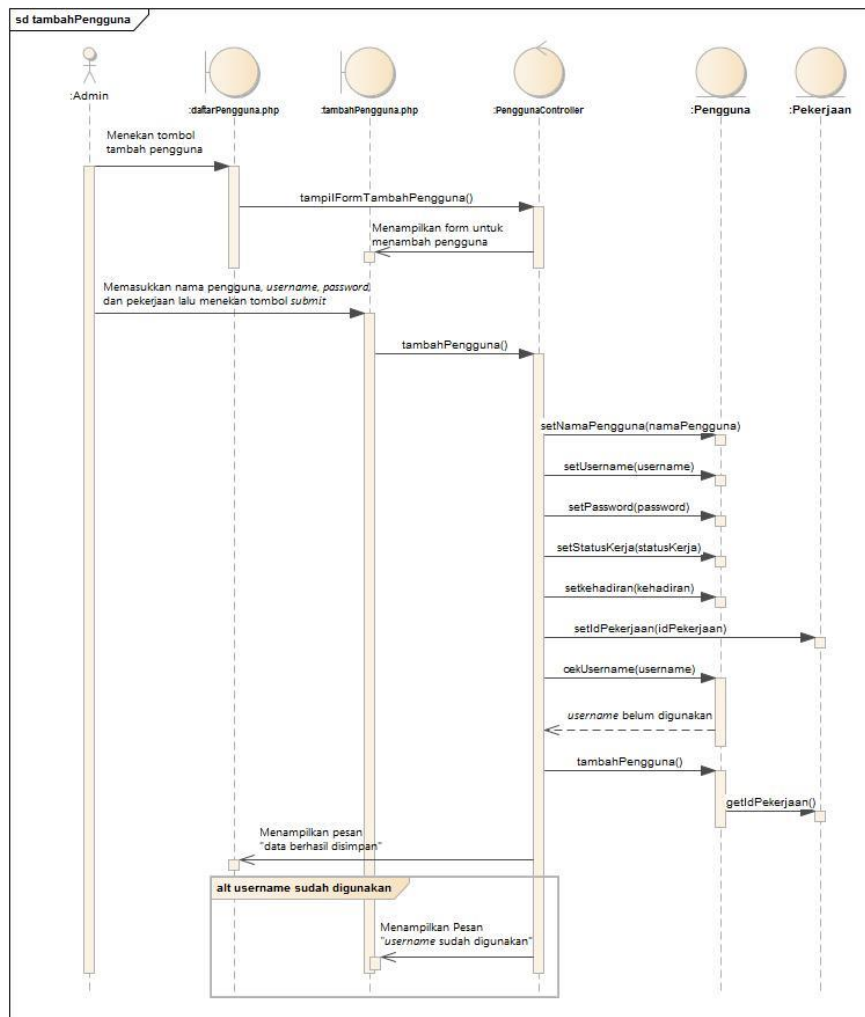
Pada *sequence diagram* menambah pasien baru, aktor yang berperan adalah petugas administrasi. Dua *boundary* yang terlibat adalah `daftarPasien.php` dan `tambahPasien.php`. Objek *controller* yang terlibat adalah `PasienController`. Objek *entity* yang terlibat adalah `Pasien`. *Sequence diagram* menambah pasien baru diilustrasikan pada Gambar 5.2.

##### ***Sequence Diagram* Menambah Obat Baru**

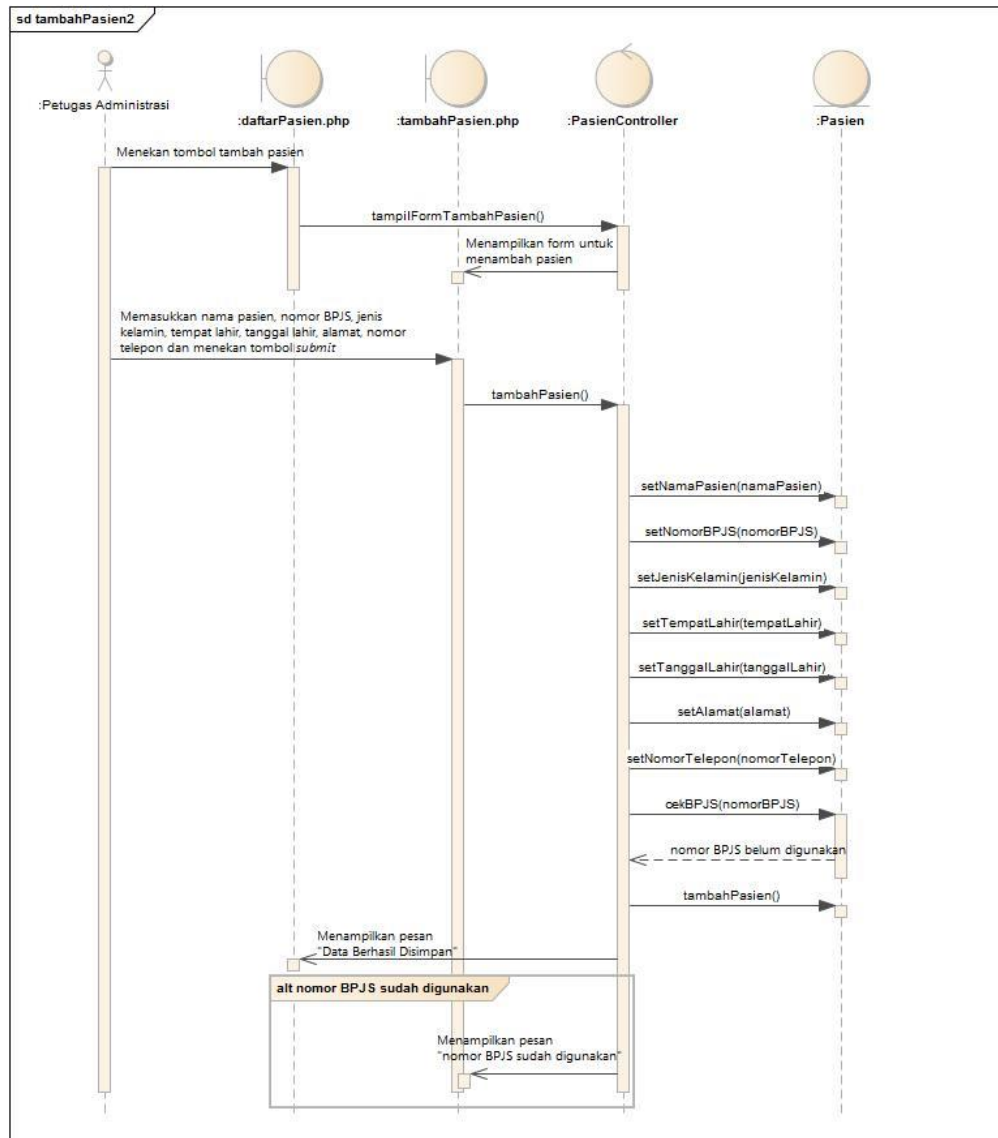
Pada *sequence diagram* menambah obat baru, aktor yang berperan adalah apoteker. Dua *boundary* yang terlibat adalah `daftarObat.php` dan `tambahObat.php`. Objek *controller* yang terlibat adalah `ObatController`. Objek *entity* yang terlibat adalah `Obat`. *Sequence diagram* menambah obat baru diilustrasikan pada Gambar 5.3.

### 5.1.1.2 Pemodelan *Class Diagram*

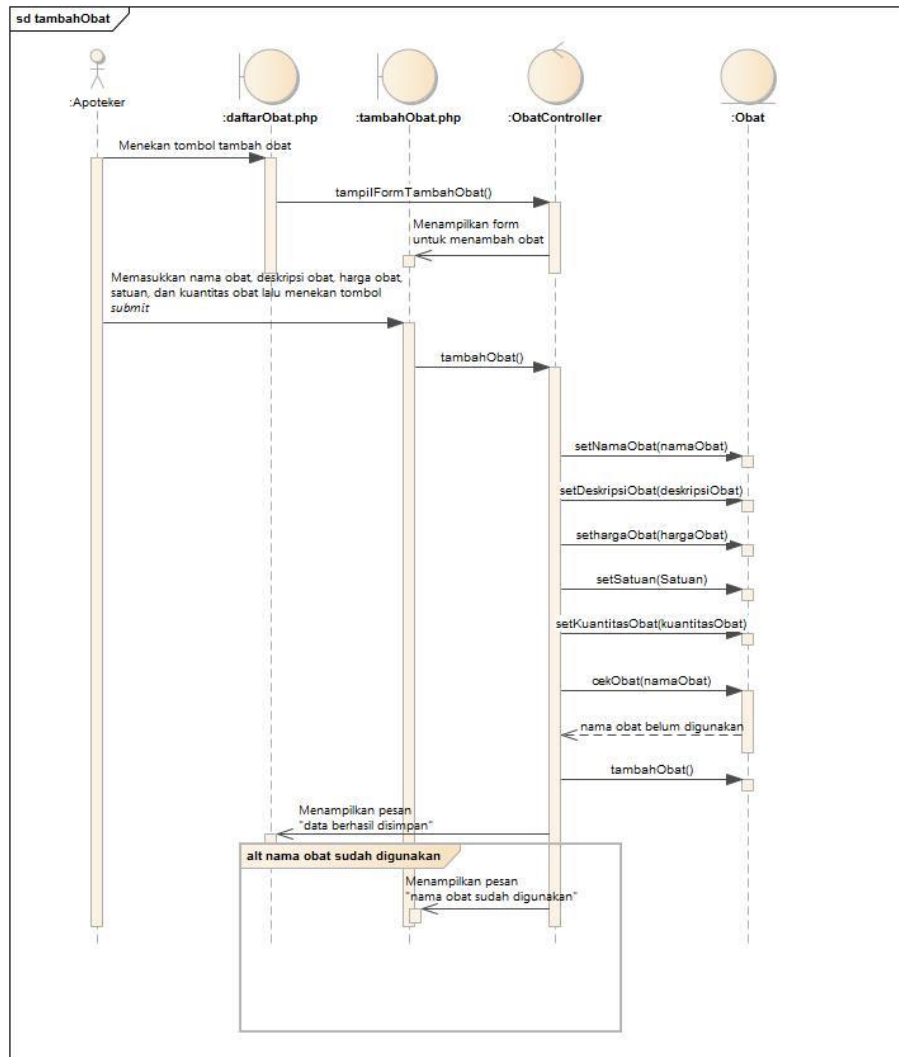
Tahap pemodelan *class diagram* menggambarkan klas-klas yang membentuk sistem dan relasi antarklas tersebut. Pemodelan *class diagram* yang ditampilkan adalah pemodelan *class diagram* umum, pemodelan *class diagram controller*, dan pemodelan *class diagram entity*. Pemodelan *class diagram* umum menggambarkan klas-klas yang akan membentuk sistem dan relasi antarklas tersebut. Klas-klas tersebut terdiri dari *boundary*, *controller*, dan *entity*. Klas-klas *boundary* memiliki relasi asosiasi terhadap klas-klas *controller*. Klas-klas *controller* memiliki relasi asosiasi terhadap klas-klas *entity*. Pemodelan *class diagram controller* menggambarkan klas-klas *controller* yang terdapat pada pemodelan *class diagram* umum, dengan disertakan *method* klas tersebut. Pemodelan *class diagram entity* menggambarkan klas-klas *entity* yang terdapat pada pemodelan *class diagram* umum, dengan disertakan atribut, operasi, dan relasi antarklas. Pemodelan *class diagram* umum diilustrasikan pada Gambar 5.4. Pemodelan *class diagram controller* diilustrasikan pada Gambar 5.5. Pemodelan *class diagram entity* diilustrasikan pada Gambar 5.6.



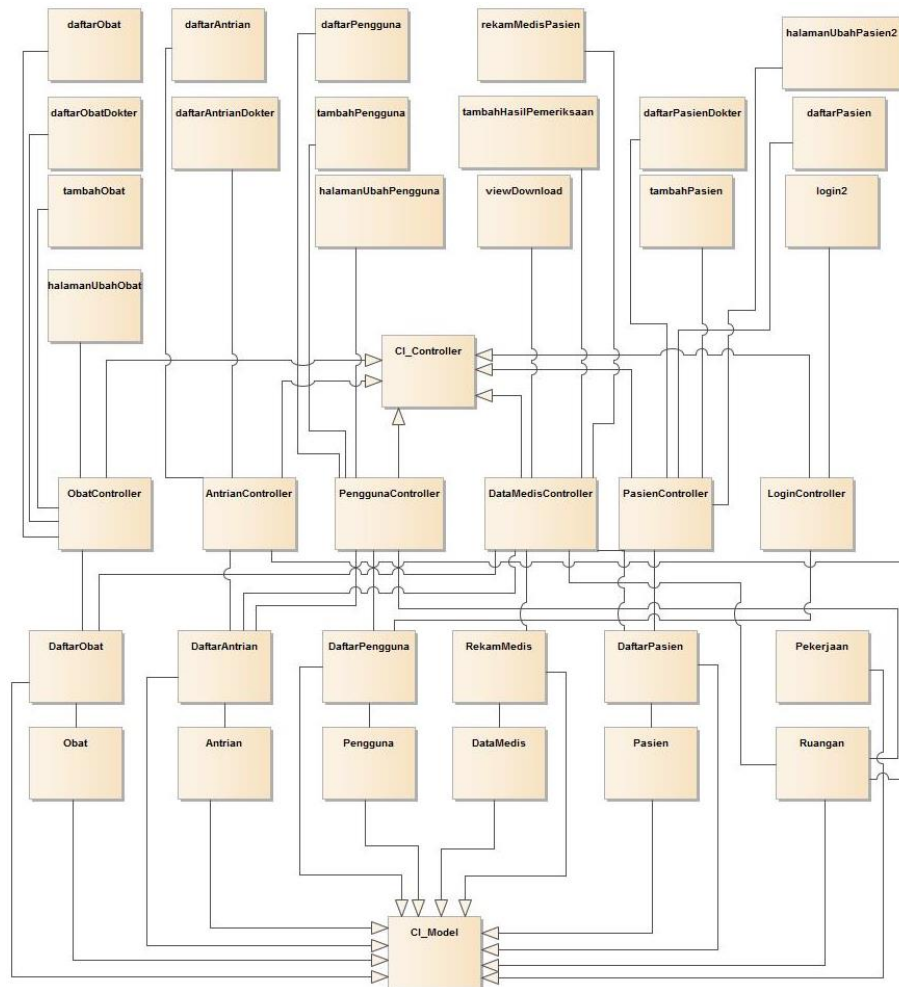
Gambar 5.1 *Sequence diagram* Menambah Pengguna



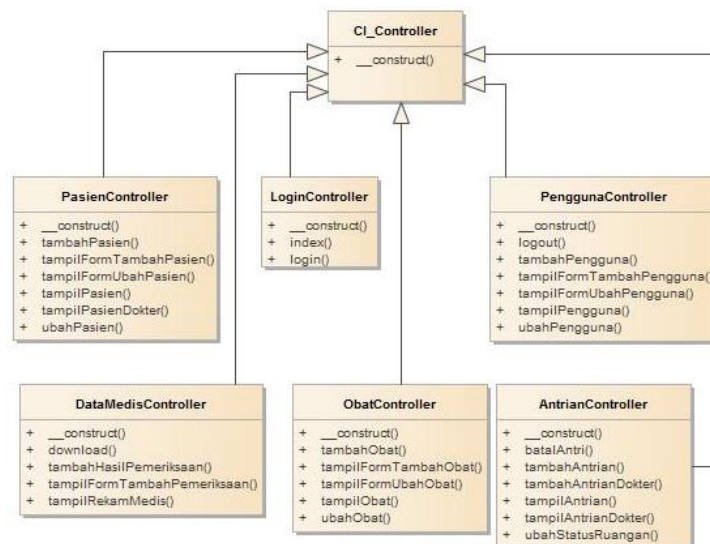
Gambar 5.2 Sequence diagram Menambah Pasien Baru



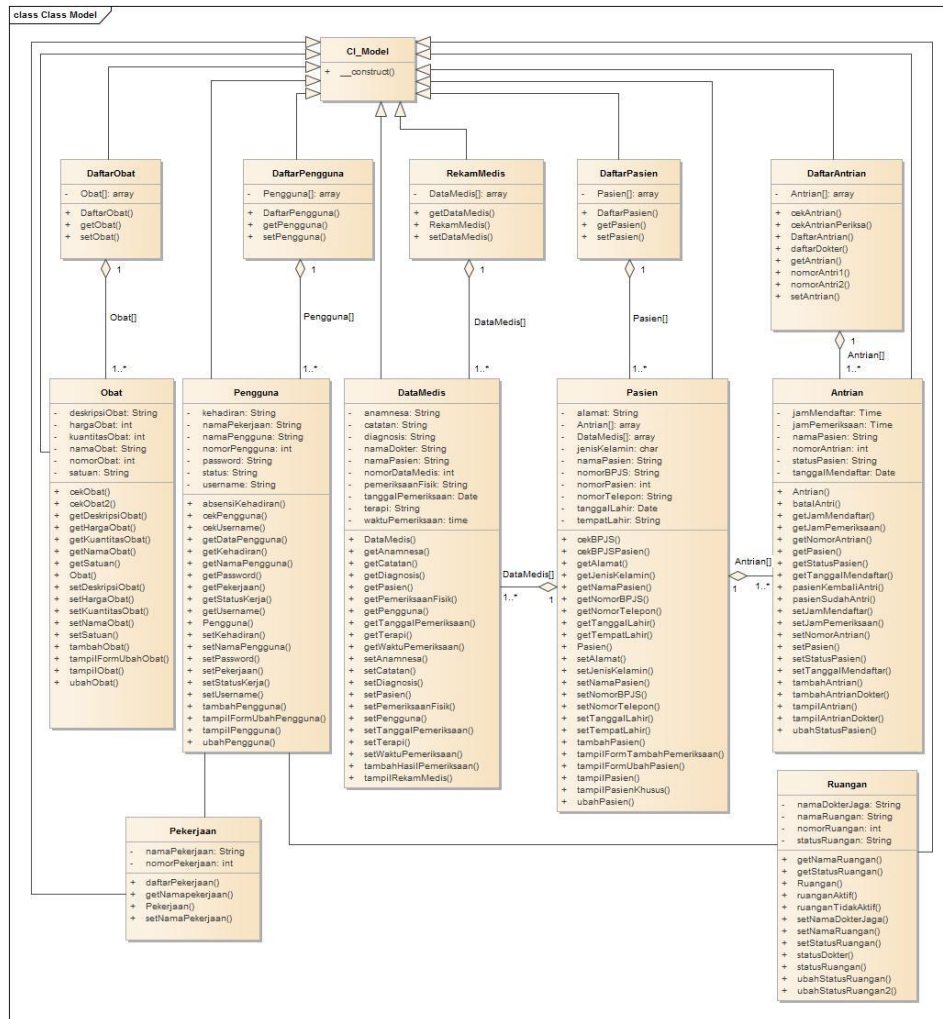
**Gambar 5.3 Sequence diagram Menambah Obat Baru**



**Gambar 5.4** Pemodelan *class diagram* umum



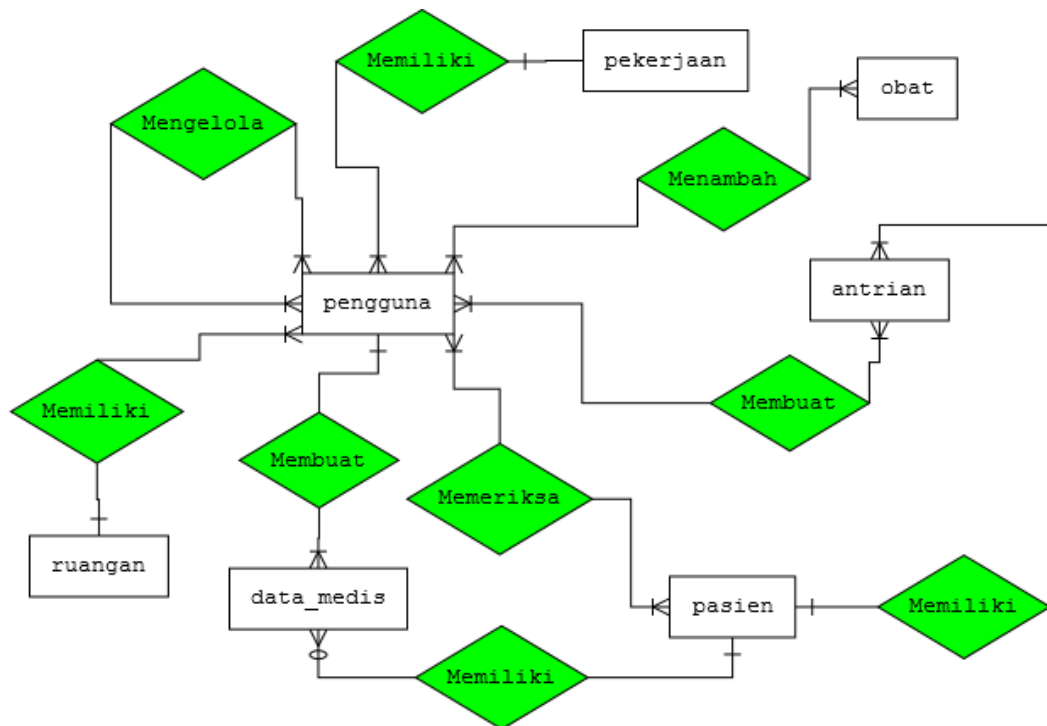
**Gambar 5.5** Pemodelan *class diagram* controller



Gambar 5.6 Pemodelan *class diagram entity*

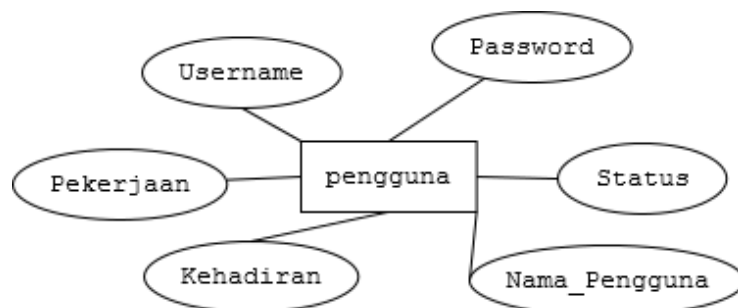
### 5.1.2 Perancangan Data

Perancangan data dimodelkan dengan CDM dan PDM. Pemodelan CDM memiliki 7 entitas, yaitu pengguna, pekerjaan, obat, pasien, antrian, ruangan, dan data\_medis. Gambar 5.7 menunjukkan pemodelan CDM dengan entitas dan relasi tanpa menyertakan atribut. Perancangan PDM dilakukan dengan mentransformasikan entitas dan atribut yang terdapat pada CDM. Entitas-entitas pada CDM akan dirubah menjadi tabel dan atributnya menjadi *column*. Perancangan PDM menghasilkan 7 tabel, yaitu pengguna, pekerjaan, obat, pasien, antrian, ruangan, dan data\_medis. Pemodelan PDM dapat dilihat pada Gambar 5.15. Proses normalisasi tidak dilakukan dikarenakan struktur tabel pada PDM telah normal dan tidak ada redundansi data.



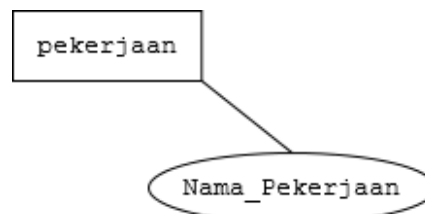
**Gambar 5.7 Pemodelan CDM**

Gambar 5.8 menunjukkan detail dari entitas *pengguna* dengan menampilkan atribut yang dimiliki oleh entitas *pengguna*, yaitu *Username*, *Password*, *Pekerjaan*, *Kehadiran*, *Status*, dan *Nama\_Pengguna*.



**Gambar 5.8 Detail entitas pengguna**

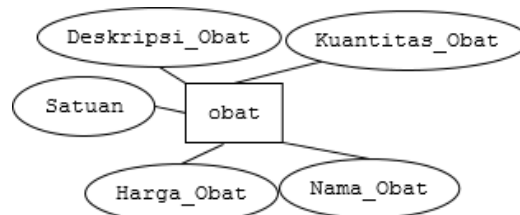
Gambar 5.9 menunjukkan detail dari entitas *pekerjaan* dengan menampilkan atribut yang dimiliki oleh entitas *pekerjaan*, yaitu *Nama\_Pekerjaan*.



**Gambar 5.9 Detail entitas pekerjaan**

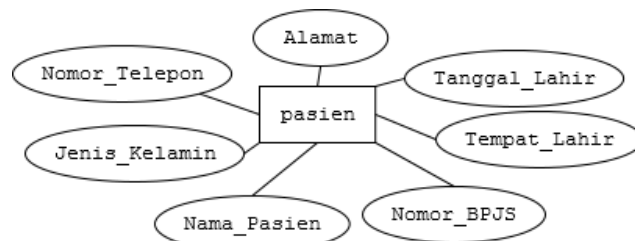


Gambar 5.10 menunjukkan detail dari entitas obat dengan menampilkan atribut yang dimiliki oleh entitas obat, yaitu Nama\_Obat, Deskripsi\_Obat, Kuantitas\_Obat, Satuan, dan Harga\_Obat.



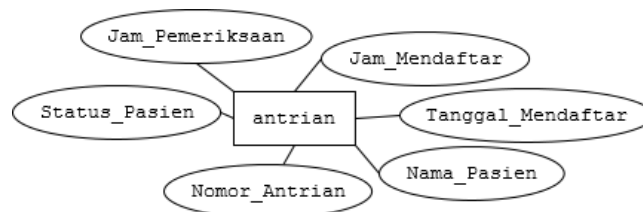
**Gambar 5.10 Detail entitas obat**

Gambar 5.11 menunjukkan detail dari entitas pasien dengan menampilkan atribut yang dimiliki oleh entitas pasien, yaitu Nama\_Pasien, Nomor\_BPJS, Jenis\_Kelamin, Tanggal\_Lahir, Tempat\_Lahir, Alamat, dan Nomor\_Telepon.



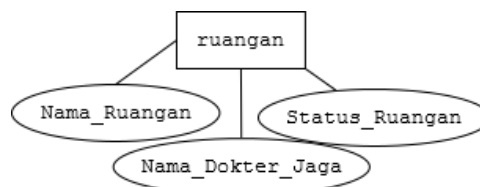
**Gambar 5.11 Detail entitas pasien**

Gambar 5.12 menunjukkan detail dari entitas antrian dengan menampilkan atribut yang dimiliki oleh entitas antrian, yaitu Nomor\_Antrian, Nama\_Pasien, Tanggal\_Mendaftar, Jam\_Mendaftar, Jam\_Pemeriksaan, dan Status\_Pasien.



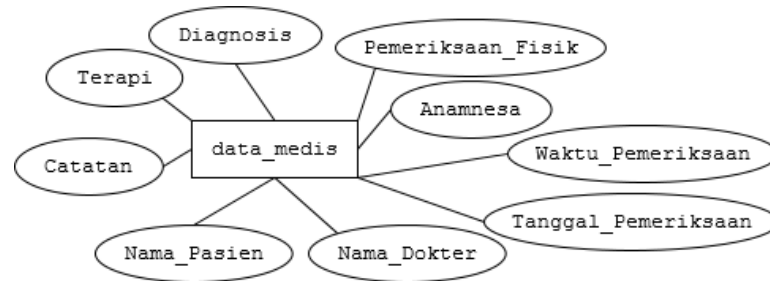
**Gambar 5.12 Detail entitas antrian**

Gambar 5.13 menunjukkan detail dari entitas ruangan dengan menampilkan atribut yang dimiliki oleh entitas ruangan, yaitu Nama\_Ruangan, Status\_Ruangan, dan Nama\_Dokter\_Jaga.

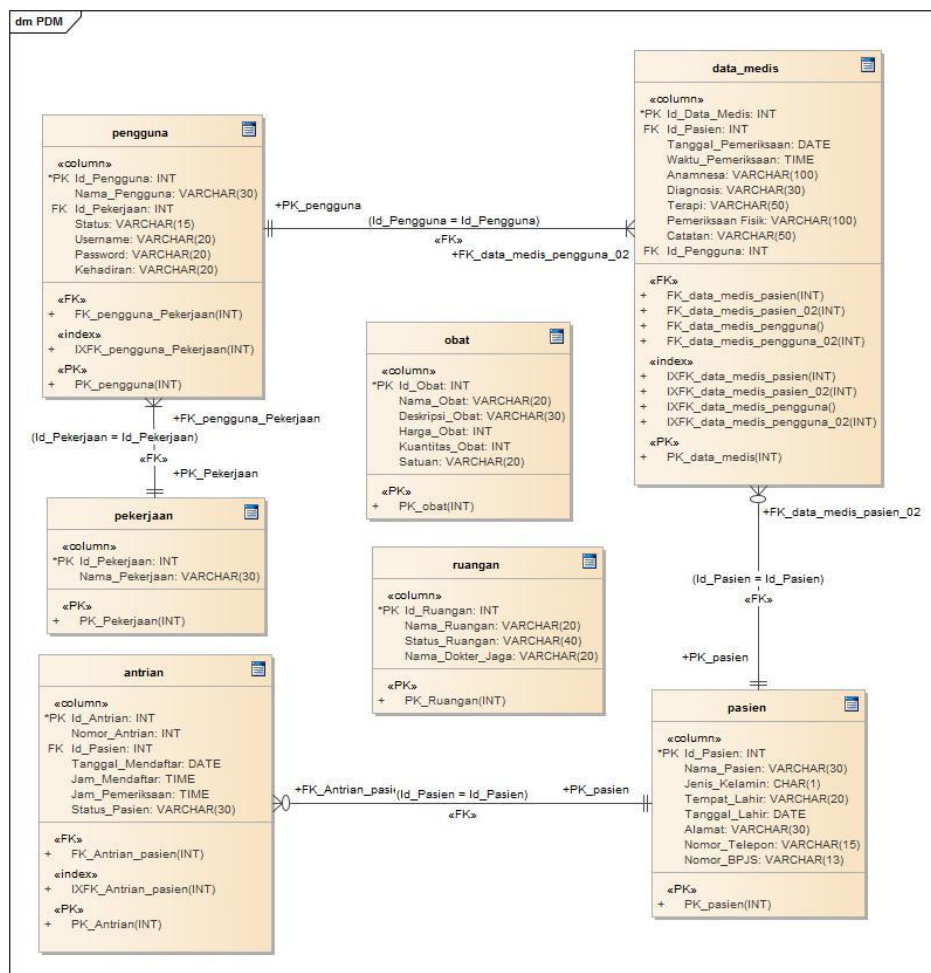


**Gambar 5.13 Detail entitas ruangan**

Gambar 5.14 menunjukkan detail dari entitas data medis dengan menampilkan atribut yang dimiliki oleh entitas data\_medis, yaitu Nama\_Pasien, Nama\_Dokter, Tanggal\_Pemeriksaan, Waktu\_Pemeriksaan, Anamnesa, Pemeriksaan\_Fisik, Diagnosis, Terapi, dan Catatan.



Gambar 5.14 Detail entitas data\_medis



Gambar 5.15 Pemodelan PDM

### 5.1.3 Perancangan Komponen

Perancangan komponen menggambarkan tentang rincian dari setiap sub-sistem komponen perangkat lunak. Perancangan komponen harus mendefinisikan struktur data untuk semua objek data dan rincian algoritme untuk proses yang terjadi di dalam komponen. Berikut ini adalah beberapa rancangan komponen yang terdapat di kelas *Entity*.

#### 5.1.3.1 Perancangan Komponen *Method* tambahPasien() klas Pasien

Tabel 5.1 merupakan *pseudocode* dari *method* tambahPasien() klas Pasien.

**Tabel 5.1 *Pseudocode method* tambahPasien() klas Pasien**

No	<i>Pseudocode</i>
1	START
2	data = array( 3     'Nama_Pasien' = namaPasien 4     'Nomor_BPJS' = nomorBPJS 5     'Jenis_Kelamin' = jenisKelamin 6     'Tempat_Lahir' = tempatLahir 7     'Tanggal_Lahir' = tanggalLahir 8     'Alamat' = alamat 9     'Nomor_Telepon' = nomorTelepon) 10    Run query insert with data = data 11    END

#### 5.1.3.2 Perancangan Komponen *Method* tambahPegguna() klas Pegguna

Tabel 5.2 merupakan *pseudocode* dari *method* tambahPegguna() klas Pegguna.

**Tabel 5.2 *Pseudocode method* tambahPegguna() klas Pegguna**

No	<i>Pseudocode</i>
1	START
2	data = array( 3     'Nama_Pegguna' = namaPegguna, 4     'Id_Pekerjaan' = getIdPekerjaan() from object Pekerjaan, 5     'Username' = username, 6     'Password' = password, 7     'Status' = statusKerja, 8     'Kehadiran' = kehadiran) 9    Run query insert with data = data 10    END

#### 5.1.3.3 Perancangan Komponen *Method* tambahObat() klas Obat

Tabel 5.3 merupakan *pseudocode* dari *method* tambahObat() klas Obat.

**Tabel 5.3 *Pseudocode method* tambahObat() klas Obat**

No	<i>Pseudocode</i>
----	-------------------

1	<i>START</i>
2	<i>data = array(</i>
3	<i>'Nama_Obat' = namaObat,</i>
4	<i>'Deskripsi_Obat' = deskripsiObat,</i>
5	<i>'Harga_Obat' = hargaObat,</i>
6	<i>'Satuan' = satuan,</i>
7	<i>'Kuantitas_Obat' = kuantitasObat)</i>
8	<i>Run query insert with data = data</i>
9	<i>END</i>

#### 5.1.4 Perancangan Antarmuka

Tahap ini merupakan tahap merancang tampilan dari sistem yang akan dibuat. Tiga buah sampel antarmuka yang akan ditampilkan pada tahap ini adalah antarmuka halaman menambah pasien, halaman menambah pengguna, dan halaman menambah obat baru.

##### 5.1.4.1 Perancangan Antarmuka Halaman Menambah Pasien

Perancangan antarmuka halaman menambah pasien dapat dilihat pada Gambar 5.16 dan penjelasan detail dari perancangan antarmuka tersebut dapat dilihat pada Tabel 5.4.

**Gambar 5.16 Perancangan antarmuka Halaman Menambah Pasien**

**Tabel 5.4 Penjelasan antarmuka Halaman Menambah Pasien**

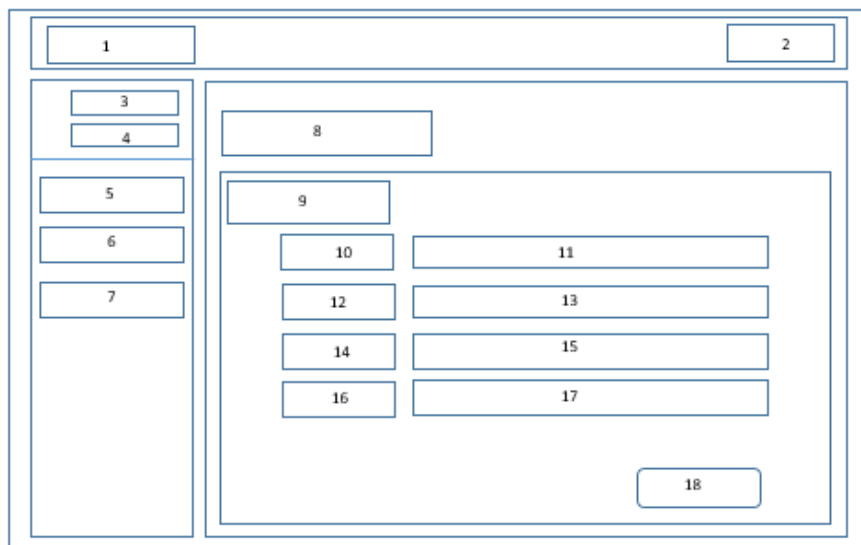
No	Nama Objek	Tipe	Keterangan
1	Judul Aplikasi	<i>Text</i>	Menampilkan judul aplikasi
2	Logout	<i>Logo Button</i>	Menampilkan logo untuk keluar dari sistem

3	<i>Username</i>	<i>Text</i>	Menampilkan <i>username</i> dari pengguna yang telah masuk ke dalam sistem
4	Pekerjaan	<i>Text</i>	Menampilkan pekerjaan pengguna yang telah masuk ke dalam sistem
5	Lihat Pasien	<i>Menu</i>	Untuk menampilkan daftar pasien yang terdapat pada <i>database</i>
6	Lihat Obat	<i>Menu</i>	Untuk menampilkan daftar obat yang terdapat pada <i>database</i>
7	Lihat Antrian	<i>Menu</i>	Untuk menampilkan daftar antrian yang sedang berlangsung pada sistem
8	Nama Aksi	<i>Page Header</i>	Menampilkan aksi apa yang sedang dilakukan oleh petugas administrasi
9	Nama Form	<i>Text</i>	Menampilkan nama pasien yang hasil pemeriksaannya akan dimasukkan ke dalam <i>database</i>
10	Nama Pasien	<i>Text label</i>	Menampilkan label nama pasien
11	Kolom Nama Pasien	<i>Input area</i>	Menampilkan kolom untuk mengisi nama pasien
12	Nomor BPJS	<i>Text label</i>	Menampilkan label nomor BPJS
13	Kolom Nomor BPJS	<i>Input area</i>	Menampilkan kolom untuk mengisi nomor BPJS pasien
14	Jenis Kelamin	<i>Text label</i>	Menampilkan label jenis kelamin

15	Kolom Jenis Kelamin	<i>Input area</i>	Menampilkan kolom untuk mengisi jenis kelamin pasien
16	Tempat Lahir	<i>Text label</i>	Menampilkan label tempat lahir
17	Kolom Tempat Lahir	<i>Input area</i>	Menampilkan kolom untuk mengisi tempat lahir pasien
18	Tanggal Lahir	<i>Text label</i>	Menampilkan label tanggal lahir
19	Kolom Tanggal Lahir	<i>Input area</i>	Menampilkan kolom untuk mengisi tanggal lahir pasien
20	Alamat	<i>Text label</i>	Menampilkan label alamat
21	Kolom Alamat	<i>Input area</i>	Menampilkan kolom untuk mengisi alamat pasien
22	Nomor Telepon	<i>Text label</i>	Menampilkan label nomor telepon
23	Kolom Nomor Telepon	<i>Input area</i>	Menampilkan kolom untuk mengisi nomor telepon pasien
24	<i>Submit</i>	<i>button</i>	Untuk melakukan penyimpanan pasien baru

#### 5.1.4.2 Perancangan Antarmuka Halaman Menambah Pengguna

Perancangan antarmuka halaman menambah pengguna dapat dilihat pada Gambar 5.17 dan penjelasan detail dari perancangan antarmuka tersebut dapat dilihat pada Tabel 5.5.



**Gambar 5.17** Perancangan antarmuka Halaman Menambah Pengguna

**Tabel 5.5** Penjelasan antarmuka Halaman Menambah Pengguna

No	Nama Objek	Tipe	Keterangan
1	Judul Aplikasi	<i>Text</i>	Menampilkan judul aplikasi
2	Logout	<i>Logo Button</i>	Menampilkan logo untuk keluar dari sistem
3	<i>Username</i>	<i>Text</i>	Menampilkan <i>username</i> dari pengguna yang telah masuk ke dalam sistem
4	Pekerjaan	<i>Text</i>	Menampilkan pekerjaan pengguna yang telah masuk ke dalam sistem
5	Lihat Pengguna	<i>Menu</i>	Untuk menampilkan daftar pengguna yang terdapat pada <i>database</i>
6	Lihat Pasien	<i>Menu</i>	Untuk menampilkan daftar pasien yang terdapat pada <i>database</i>
7	Lihat Antrian	<i>Menu</i>	Untuk menampilkan daftar antrian yang sedang berlangsung pada sistem
8	Nama Aksi	<i>Page Header</i>	Menampilkan aksi apa yang sedang dilakukan oleh petugas administrasi
9	Nama Form	<i>text</i>	Menampilkan nama form

10	Nama Pengguna	<i>Text label</i>	Menampilkan label nama pengguna
11	Kolom nama pengguna	<i>Input area</i>	Menampilkan kolom untuk mengisi nama pengguna
12	<i>Username</i>	<i>Text label</i>	Menampilkan label <i>username</i>
13	Kolom <i>username</i>	<i>Input area</i>	Menampilkan kolom untuk mengisi <i>username</i>
14	<i>password</i>	<i>Text label</i>	Menampilkan label tempat <i>password</i>
15	Kolom <i>password</i>	<i>Input area</i>	Menampilkan kolom untuk mengisi <i>password</i>
16	Status	<i>Text label</i>	Menampilkan label status
17	Kolom Status	<i>Input area</i>	Menampilkan kolom untuk mengisi status pengguna
18	<i>submit</i>	<i>button</i>	Untuk melakukan penyimpanan penambahan pengguna

#### 5.1.4.3 Perancangan Antarmuka Halaman Menambah Obat Baru

Perancangan antarmuka halaman menambah obat baru dapat dilihat pada Gambar 5.18 dan penjelasan detail dari perancangan antarmuka tersebut dapat dilihat pada Tabel 5.6.

**Gambar 5.18 Perancangan antarmuka Halaman Menambah Obat Baru**



**Tabel 5.6 Penjelasan antarmuka Halaman Menambah Obat Baru**

No	Nama Objek	Tipe	Keterangan
1	Judul Aplikasi	<i>Text</i>	Menampilkan judul aplikasi
2	Logout	<i>Logo Button</i>	Menampilkan logo untuk keluar dari sistem
3	<i>Username</i>	<i>Text</i>	Menampilkan <i>username</i> dari pengguna yang telah masuk ke dalam sistem
4	Pekerjaan	<i>Text</i>	Menampilkan pekerjaan pengguna yang telah masuk ke dalam sistem
5	Lihat Obat	<i>Menu</i>	Untuk menampilkan daftar pengguna yang terdapat pada <i>database</i>
6	Nama Aksi	<i>Page Header</i>	Menampilkan aksi apa yang sedang dilakukan oleh petugas administrasi
7	Nama Form	<i>text</i>	Menampilkan nama form
8	Nama Obat	<i>Text label</i>	Menampilkan label nama obat
9	Kolom nama obat	<i>Input area</i>	Menampilkan kolom untuk mengisi nama obat
10	Deskripsi obat	<i>Text label</i>	Menampilkan label deskripsi obat
11	Kolom deskripsi obat	<i>Input area</i>	Menampilkan kolom untuk mengisi deskripsi obat
12	Harga obat	<i>Text label</i>	Menampilkan label harga obat
13	Kolom harga obat	<i>Input area</i>	Menampilkan kolom untuk mengisi harga obat
14	Satuan Obat	<i>Text label</i>	Menampilkan label satuan obat
15	Kolom Satuan Obat	<i>Input area</i>	Menampilkan kolom untuk mengisi satuan obat
16	Kuantitas obat	<i>Text label</i>	Menampilkan label kuantitas obat
17	Kolom kuantitas obat	<i>Input area</i>	Menampilkan kolom untuk mengisi kuantitas obat

18	<i>Cancel</i>	<i>button</i>	Untuk berhenti mengubah data obat dan kembali ke halaman daftar obat
19	<i>Submit</i>	<i>button</i>	Untuk melakukan penyimpanan perubahan obat

## 5.2 Implementasi

Implementasi dilakukan setelah melakukan tahap perancangan. Implementasi dilakukan berdasarkan hasil dari tahap perancangan. Tahap implementasi ini akan menjelaskan tentang spesifikasi sistem yang digunakan dalam mengembangkan sistem, batasan implementasi, implementasi data, implementasi kode program, dan implementasi antarmuka.

### 5.2.1 Spesifikasi Sistem

Spesifikasi sistem yang digunakan dalam membangun sistem ini terbagi menjadi spesifikasi perangkat keras, perangkat lunak, dan sistem operasi.

#### 5.2.1.1 Spesifikasi Perangkat Keras

Pengembangan sistem pengelolaan rekam medis pasien berbasis web menggunakan perangkat keras berupa komputer dengan spesifikasi yang dapat dilihat pada Tabel 5.7.

**Tabel 5.7 Spesifikasi perangkat keras**

Nama	Spesifikasi
<i>System Model</i>	ASUS X455LB
<i>processor</i>	Intel(R) Core(TM) i7-5500U CPU @2.40GHz (4 CPUs)
Memory	8,00 GB
Display	Intel HD Graphic 5500
Render Name	NVIDIA GeForce 940M
Harddisk	TOSHIBA MQ01ABD100 1,00 TB

#### 5.2.1.2 Spesifikasi Perangkat Lunak

Pengembangan sistem pengelolaan rekam medis pasien berbasis web ini menggunakan perangkat lunak dengan spesifikasi yang dapat dilihat pada Tabel 5.8.

**Tabel 5.8 Spesifikasi perangkat lunak**

Nama Komponen	Spesifikasi
Editor Perancangan	Enterprise Architect
Editor Pemrograman	Sublime Text 3 Build 3126
Framework dan Bahasa Pemrograman	PHP, CSS, javaScript, Codeigniter
DBMS	MySQL
Database server	Phpmyadmin
Aplikasi Pendukung Sistem	XAMPP
Browser	1. Mozilla Firefox 60.0.1 (64-bit) version 2. Google Chrome 67.0.3 (64-bit) version

#### 5.2.1.3 Spesifikasi Sistem Operasi

Pengembangan sistem pengelolaan rekam medis pasien berbasis web ini menggunakan sistem operasi dengan spesifikasi yang dapat dilihat pada Tabel 5.9.

**Tabel 5.9 Spesifikasi sistem operasi**

Nama Komponen	Spesifikasi
Sistem Operasi	Windows 8.1 Single Language 64-bit (6.3, build 9600)
Language	Bahasa Indonesia (Regional Setting: Bahasa Indonesia)

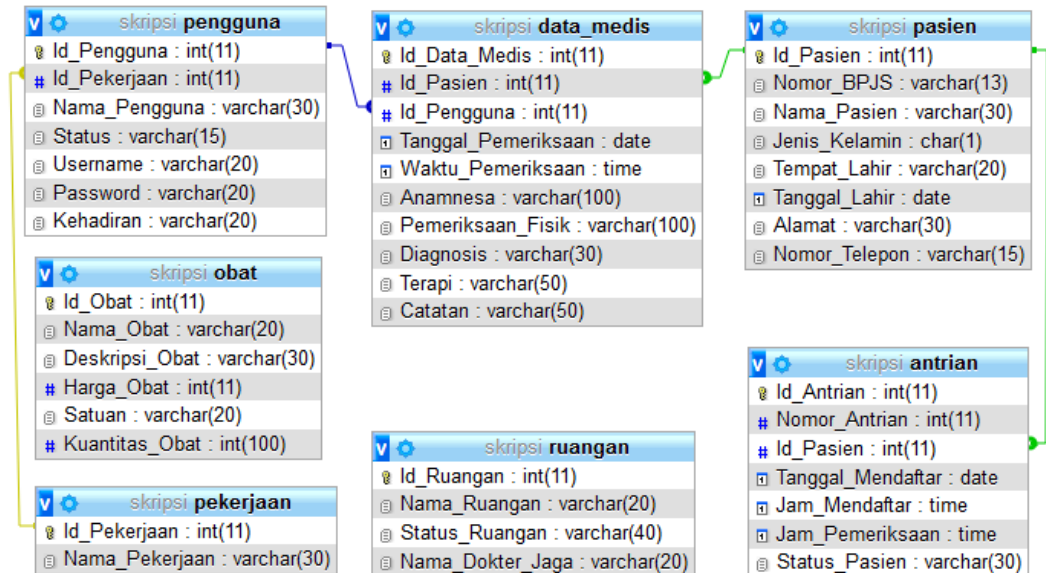
#### 5.2.2 Batasan Implementasi

Terdapat beberapa batasan pada proses implementasi sistem ini, yaitu:

1. Bahasa pemrograman yang digunakan adalah PHP dan JavaScript yang diimplementasikan pada *framework* Codeigniter.
2. Database yang digunakan adalah MySQL.
3. Antarmuka sistem dibuat menggunakan HTML, JavaScript, dan CSS.

### 5.2.3 Implementasi Data

Implementasi data menerapkan perancangan data yang sebelumnya telah dibuat. Implementasi data pada basis data sistem memiliki 7 tabel, yaitu pengguna, data\_medis, pasien, obat, pekerjaan, ruangan, dan antrian. Implementasi data diilustrasikan pada Gambar 5.19.



Gambar 5.19 Implementasi data

### 5.2.4 Implementasi Kode Program

Pada tahap ini, implementasi kode program dibuat berdasarkan perancangan komponen yang telah dilakukan sebelumnya. Algoritme yang sebelumnya dibuat dalam bentuk *pseudocode* akan dirubah ke dalam bentuk bahasa pemrograman. Algoritme yang akan ditulis implementasinya yaitu menambah pasien, menambah pengguna, dan menambah obat.

#### 5.2.4.1 Implementasi Kode Program Method tambahPasien() klas Pasien

Tabel 5.10 berikut merupakan implementasi kode program dari method “tambahPasien” klas “Pasien”.

Tabel 5.10 Implementasi *method* tambahPasien() klas Pasien

No	Source code
1	<code>\$data = array(</code>
2	<code>'Nama_Pasien' =&gt;\$this-&gt;namaPasien,</code>
3	<code>'Nomor_BPJS' =&gt;\$this-&gt;nomorBPJS,</code>
4	<code>'Jenis_Kelamin' =&gt;\$this-&gt;jenisKelamin,</code>
5	<code>'Tempat_Lahir' =&gt;\$this-&gt;tempatLahir,</code>
6	<code>'Tanggal_Lahir' =&gt;\$this-&gt;tanggalLahir,</code>
7	<code>'Alamat' =&gt;\$this-&gt;alamat,</code>
8	<code>'Nomor_Telepon' =&gt;\$this-&gt;nomorTelepon);</code>
9	<code>\$this-&gt;db-&gt;insert('pasien',\$data);</code>

#### 5.2.4.2 Implementasi Kode Program Method *tambahPengguna()* klas *Pengguna*

Tabel 5.11 berikut merupakan implementasi kode program dari method “tambahPengguna” klas “Pengguna”.

**Tabel 5.11 Implementasi *method* tambahPengguna klas Pengguna**

No	Source code
1	<code>\$data = array(</code>
2	<code>'Nama_Pengguna'=&gt;\$this-&gt;namaPengguna,</code>
3	<code>'Id_Pekerjaan' =&gt;\$this-&gt;pekerjaan-&gt;getIdPekerjaan(),</code>
4	<code>'Username' =&gt;\$this-&gt;username,</code>
5	<code>'Password' =&gt;\$this-&gt;password,</code>
6	<code>'Status' =&gt;\$this-&gt;statusKerja,</code>
7	<code>'Kehadiran' =&gt;\$this-&gt;kehadiran);</code>
8	<code>\$this-&gt;db-&gt;insert('pengguna',\$data);</code>

#### 5.2.4.3 Implementasi Kode Program Method *tambahObat()* klas *Obat*

Tabel 5.12 berikut merupakan implementasi kode program dari method “tambahObat” klas “Obat”.

**Tabel 5.12 Implementasi *method* tambahObat() klas Obat**

No	Sourcecode
1	<code>\$data = array(</code>
2	<code>'Nama_Obat'=&gt;\$this-&gt;namaObat,</code>
3	<code>'Deskripsi_Obat'=&gt;\$this-&gt;deskripsiObat,</code>
4	<code>'Harga_Obat'=&gt;\$this-&gt;hargaObat,</code>
5	<code>'Satuan'=&gt;\$this-&gt;satuan,</code>
6	<code>'Kuantitas_Obat'=&gt;\$this-&gt;kuantitasObat);</code>
7	<code>\$this-&gt;db-&gt;insert('obat',\$data);</code>

### 5.2.5 Implementasi Antarmuka

Implementasi antarmuka dilakukan berdasarkan perancangan antarmuka yang sebelumnya telah dilakukan. Berikut adalah beberapa antarmuka dari halaman-halaman yang ada pada sistem pengelolaan rekam medis pasien berbasis web.

#### 5.2.5.1 Implementasi Antarmuka Halaman Menambah Pasien

Gambar 5.20 merupakan implementasi antarmuka Halaman Menambah Pasien. Halaman ini memuat *field-field* yang diperlukan dalam proses penambahan pasien dan menu untuk *logout*, Lihat Pasien, dan Lihat Antrian.

#### 5.2.5.2 Implementasi Antarmuka Halaman Menambah Pengguna

Gambar 5.21 merupakan implementasi antarmuka Halaman Menambah Pengguna. Halaman ini memuat *field-field* yang diperlukan dalam proses penambahan pengguna dan menu untuk *logout* dan Lihat Pengguna.

### 5.2.5.3 Implementasi Antarmuka Halaman Menambah Obat Baru

Gambar 5.22 merupakan implementasi antarmuka Halaman Mengubah Data Obat. Halaman ini memuat *field-field* yang diperlukan dalam proses perubahan data obat dan menu untuk *logout* dan Lihat Obat.

The screenshot displays the 'Form Tambah Pasien' (Add Patient Form) in the SIPREM application. The interface features a dark header with the 'SIPREM' logo and a 'Logout' button. A sidebar on the left identifies the user as 'Helman Taufiqurrahman' (PETUGAS ADMINISTRASI) and provides navigation options: 'Lihat Halaman Utama', 'Lihat Pasien' (highlighted), and 'Lihat Antrian'. The main content area contains the form with the following fields: 'Nama Pasien' (text input), 'Nomor BPJS' (text input), 'Jenis Kelamin' (dropdown menu with 'L' selected), 'Tempat Lahir' (text input), 'Tanggal Lahir' (date picker showing 'mm / dd / yyyy'), 'Alamat' (text input), and 'Nomor Telepon' (text input). At the bottom right of the form are 'Cancel' and 'Submit' buttons.

Gambar 5.20 Implementasi antarmuka Halaman Menambah Pasien

The screenshot displays the 'Form Tambah Pengguna' (Add User Form) in the SIPREM application. The interface features a dark header with the 'SIPREM' logo and a 'Logout' button. A sidebar on the left identifies the user as 'Ismail Mahardika' (ADMIN) and provides navigation options: 'Lihat Halaman Utama' and 'Lihat Pengguna' (highlighted). The main content area contains the form with the following fields: 'Nama Pengguna' (text input with placeholder 'Your name'), 'Username' (text input with placeholder 'Your username'), 'Password' (text input with placeholder 'Your password'), and 'Pekerjaan' (dropdown menu with 'Petugas Administrasi' selected). At the bottom right of the form are 'Cancel' and 'Submit' buttons.

Gambar 5.21 Implementasi antarmuka Halaman Menambah Pengguna

SIPREM

Logout

Muhammad Sodik  
● APOTEKER

Lihat Halaman Utama

Lihat Obat

Tambah Obat

Form Tambah Obat

Nama Obat

Nama Obat

Deskripsi Obat

deskripsi obat

Harga Obat

harga obat

Satuan Obat

Satuan Obat

Kuantitas Obat

kuantitas obat

Cancel

Submit

**Gambar 5.22 Implementasi antarmuka Halaman Menambah Obat Baru**

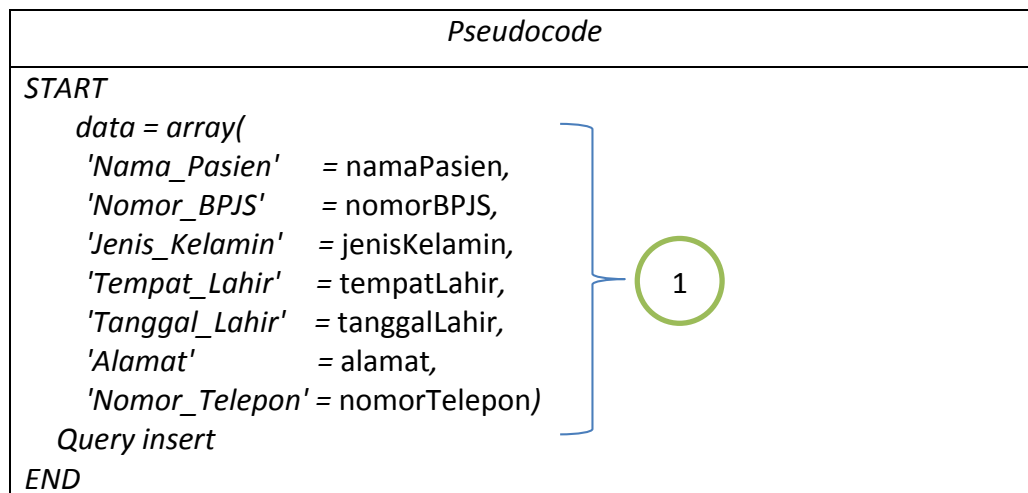
## BAB 6 PENGUJIAN

### 6.1 Pengujian Unit

Pengujian unit dilakukan pada tiga sampel, yaitu pada klas Pasien, Pengguna, dan Obat. Klas Driver digunakan untuk menjalankan unit yang akan diuji.

#### 6.1.1 Pengujian Unit *Method* tambahPasien() klas Pasien

##### 1. *pseudocode*



##### 2. *Basis Path Testing*

###### 2.1 *Flow Graph*

Gambar 6.1 berikut adalah *flow graph method* tambahPasien() dari klas Pasien.



**Gambar 6.1 *Flow graph method* tambahpasien()**

###### 2.2 *Cyclometric Complexity*

- $V(G) = \text{jumlah region} = 1$
- $V(G) = \text{jumlah edge} - \text{jumlah node} + 2 = 0 - 1 + 2 = 1$
- $V(G) = \text{jumlah predicate node} + 1 = 0 + 1 = 1$

###### 2.3 *Independent Path*

- Jalur 1 : 1




Tabel 6.1 berikut merupakan kasus uji dan hasil pengujian *method* tambahPasien klas Pasien.

**Tabel 6.1 Hasil pengujian *method* tambahPasien klas Pasien**

No.	No. Jalur	Data Pengujian	Hasil Yang Diharapkan	Hasil	Status
1	1	Klas Driver memanggil <i>method</i> tambahPasien dari klas Pasien dengan Nama pasien = "Fikri Arroisi", Nomor BPJS = "0000345763465", Jenis Kelamin = "L", Tempat Lahir = "Malang", Tanggal Lahir = 1996-09-11, Alamat = "Jalan Singosari", Nomor Telepon = "081273645382"	Memanggil <i>method</i> tambahPasien dan <i>return value true</i>	Memanggil <i>method</i> tambahPasien dan <i>return value true</i>	Valid

### 6.1.2 Pengujian Unit *Method* tambahPengguna() klas Pegguna

#### 1. *pseudocode*

<i>Pseudocode</i>	
<pre> START   data = array(     'Nama_Pegguna' = namaPegguna,     'Id_Pekerjaan' = idPekerjaan,     'Username'     = username,     'Password'     = password,     'Status'       = status,     'Kehadiran'   = kehadiran )   Query insert END </pre>	

#### 2. Basis Path Testing

##### 2.1 Flow Graph

Gambar 6.2 berikut adalah *flow graph method* tambahPegguna() dari klas Pegguna



**Gambar 6.2 Flow graph method tambahPegguna()**

## 2.2 Cyclometric Complexity (V(G))

- $V(G) = \text{jumlah region} = 1$
- $V(G) = \text{jumlah edge} - \text{jumlah node} + 2 = 0 - 1 + 2 = 1$
- $V(G) = \text{jumlah predicate node} + 1 = 0 + 1 = 1$

## 2.3 Independent Path

- Jalur 1 : 1

Tabel 6.2 berikut merupakan kasus uji dan hasil pengujian *method* tambahPengguna kelas Pengguna.

**Tabel 6.2 Hasil pengujian *method* tambahPengguna kelas Pengguna**

No.	No. Jalur	Data Pengujian	Hasil Yang Diharapkan	Hasil	Status
1	1	Klas Driver memanggil <i>method</i> tambahPengguna pada klas Pengguna dengan Nama pengguna = "Said Amin", Id pekerjaan = 3 <i>username</i> = "diasnima", <i>password</i> = "diasnima", <i>status</i> = "aktif", kehadiran = "tidak hadir"	Memanggil <i>method</i> tambahPengguna dan <i>return value true</i>	Memanggil <i>method</i> tambahPengguna dan <i>return value true</i>	Valid

## 6.1.3 Pengujian Unit *Method* tambahObat() klas Obat

### 1. pseudocode

Pseudocode	
<pre> START     data = array(         'Nama_Obat'      = namaObat,         'Deskripsi_Obat' = deskripsiObat,         'Harga_Obat'     = hargaObat,         'Satuan'         = satuan,         'Kuantitas_Obat' = kuantitasObat )     Query insert END </pre>	

## 2. Basis Path Testing

### 2.1 Flow Graph

Gambar 6.3 berikut adalah *flow graph method* tambahObat() dari klas Obat.



**Gambar 6.3 Flow graph method tambahObat()**

### 2.2 Cyclometric Complexity (V(G))

- $V(G) = \text{jumlah region} = 1$
- $V(G) = \text{jumlah edge} - \text{jumlah node} + 2 = 0 - 1 + 2 = 1$
- $V(G) = \text{jumlah predicate node} + 1 = 0 + 1 = 1$

### 2.3 Independent Path

- Jalur 1 : 1

Tabel 6.3 berikut merupakan kasus uji dan hasil pengujian *method* tambahObat klas Obat.

**Tabel 6.3 Hasil pengujian *method* tambahObat klas Obat**

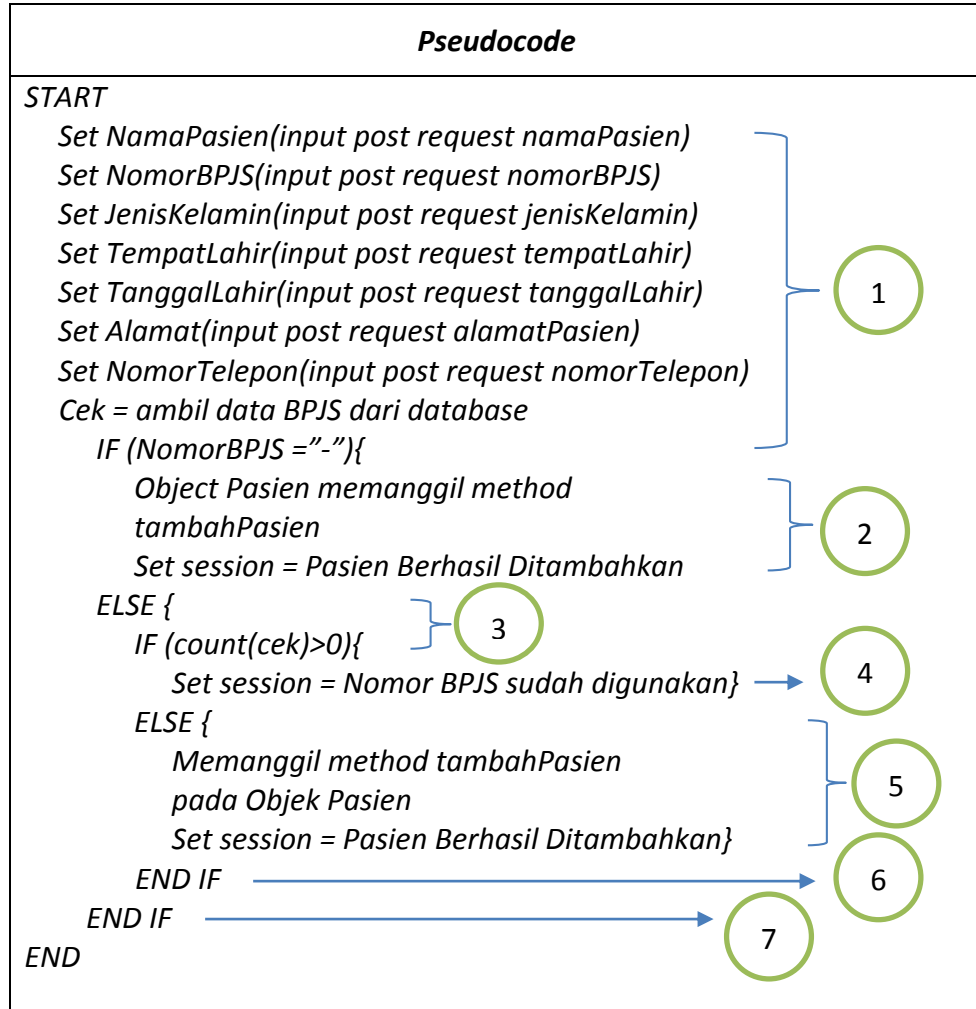
No.	No. Jalur	Data Pengujian	Hasil Yang Diharapkan	Hasil	Status
1	1	Klas Driver memanggil <i>method</i> tambahObat pada klas Obat dengan Nama obat = "Ranitidin 50mg", Deskripsi Obat = "Obat Lambung", harga obat = 6000, Satuan obat = "keping", kuantitas obat = 20	Memanggil <i>method</i> tambahObat dan <i>return value true</i>	Memanggil <i>method</i> tambahObat dan <i>return value true</i>	Valid

## 6.2 Pengujian Integrasi

Pengujian ini melibatkan klas PasienController dengan Pasien, PenggunaController dengan Pengguna dan Pekerjaan, dan ObatController dengan Obat. Pengujian dilakukan dengan menggunakan klas Driver sebagai klas yang memanggil *method* yang ada pada klas *Controller*.

### 6.2.1 Pengujian Integrasi *Method* tambahPasien() klas PasienController

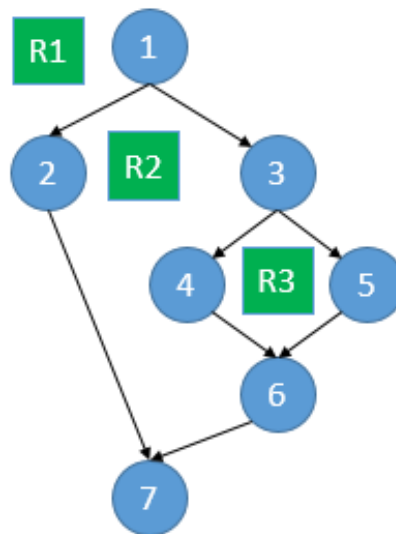
#### 1. Pseudocode



#### 2. Basis Path Testing

##### a. Flow Graph

Gambar 6.4 berikut adalah *flow graph method* tambahPasien() dari klas PasienController.



**Gambar 6.4 Flow graph** pengujian integrasi *method* tambahPasien() klas PasienController

b. Cyclomatic Complexity ( $V(G)$ )

1.  $V(G)$  = jumlah region = 3
2.  $V(G) = (\text{jumlah edge} - \text{jumlah node}) + 2 = (8 - 7) + 2 = 3$
3.  $V(G) = \text{jumlah predicate node} + 1 = 2 + 1 = 3$

c. Independent Path

- Jalur 1 : 1-2-7
- Jalur 2 : 1-3-4-6-7
- Jalur 3 : 1-3-5-6-7

Tabel 6.4 berikut merupakan kasus uji dan hasil pengujian *method* tambahPasien klas PasienController.

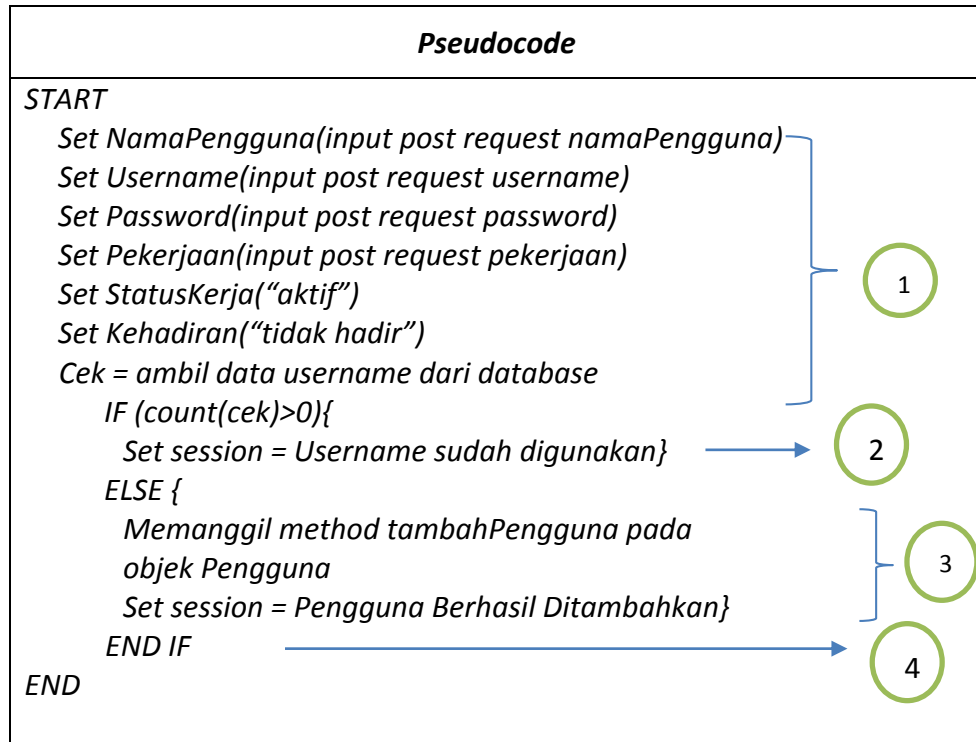
**Tabel 6.4 Hasil pengujian integrasi *method* tambahPasien()**

No. Jalur	Prosedur Uji	Expected Result	Result	Status
1	Klas Driver memanggil <i>method</i> tambahPasien() pada klas PasienController dengan set NamaPasien("Faisal"), set NomorBPJS("-"), set JenisKelamin("L"), set TempatLahir("Samarinda"), set TanggalLahir("2018-05-07"), set Alamat("Jalan Gatot Subroto"), set	Semua data pasien disimpan ke dalam <i>database</i> dan menampilkan pesan "Data berhasil disimpan"	Semua data pasien disimpan ke dalam <i>database</i> dan menampilkan pesan "Data berhasil disimpan"	Valid

	NomorTelepon("089361645 262")			
2	Klas Driver memanggil <i>method</i> tambahPasien() pada klas PasienController dengan set NamaPasien("Faisal"), set NomorBPJS("00004321432 14"), set JenisKelamin("L"), set TempatLahir("Samarinda"), set TanggalLahir("2018-05-07"), set Alamat("Jalan Gatot Subroto"), set NomorTelepon("089361645 262")	Semua data pasien gagal disimpan ke dalam <i>database</i> dan menampilkan pesan "Nomor BPJS Sudah Digunakan"	Semua data pasien gagal disimpan ke dalam <i>database</i> dan menampilkan pesan "Nomor BPJS Sudah Digunakan"	Valid
3	Klas Driver memanggil <i>method</i> tambahPasien() pada klas PasienController dengan set NamaPasien("Faisal"), set NomorBPJS("00004321432 13"), set JenisKelamin("L"), set TempatLahir("Samarinda"), set TanggalLahir("2018-05-07"), set Alamat("Jalan Gatot Subroto"), set NomorTelepon("089361645 262")	Semua data pasien disimpan ke dalam <i>database</i> dan menampilkan pesan "Data berhasil disimpan"	Semua data pasien disimpan ke dalam <i>database</i> dan menampilkan pesan "Data berhasil disimpan"	valid

## 6.2.2 Pengujian Integrasi *Method* tambahPengguna() klas PenggunaController

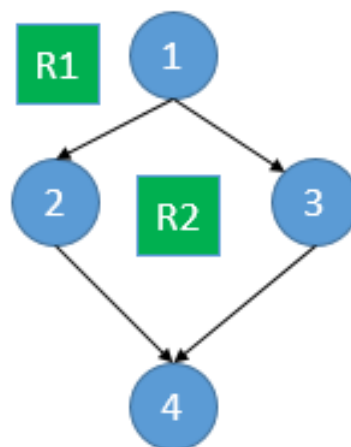
### 1. Pseudocode



### 2. Basis Path Testing

#### a. Flow Graph

Gambar 6.5 berikut adalah *flow graph method* tambahPengguna() dari klas PenggunaController.



Gambar 6.5 *Flow graph* pengujian integrasi *method* tambahPengguna() klas PenggunaController

b. *Cyclomatic Complexity* (V(G))

1.  $V(G) = \text{jumlah region} = 2$
2.  $V(G) = (\text{jumlah edge} - \text{jumlah node}) + 2 = (4 - 4) + 2 = 2$
3.  $V(G) = \text{jumlah predicate node} + 1 = 1 + 1 = 2$

c. *Independent Path*

- Jalur 1 : 1-2-4
- Jalur 2 : 1-3-4

Tabel 6.5 berikut merupakan kasus uji dan hasil pengujian *method* tambahPengguna kelas PenggunaController.

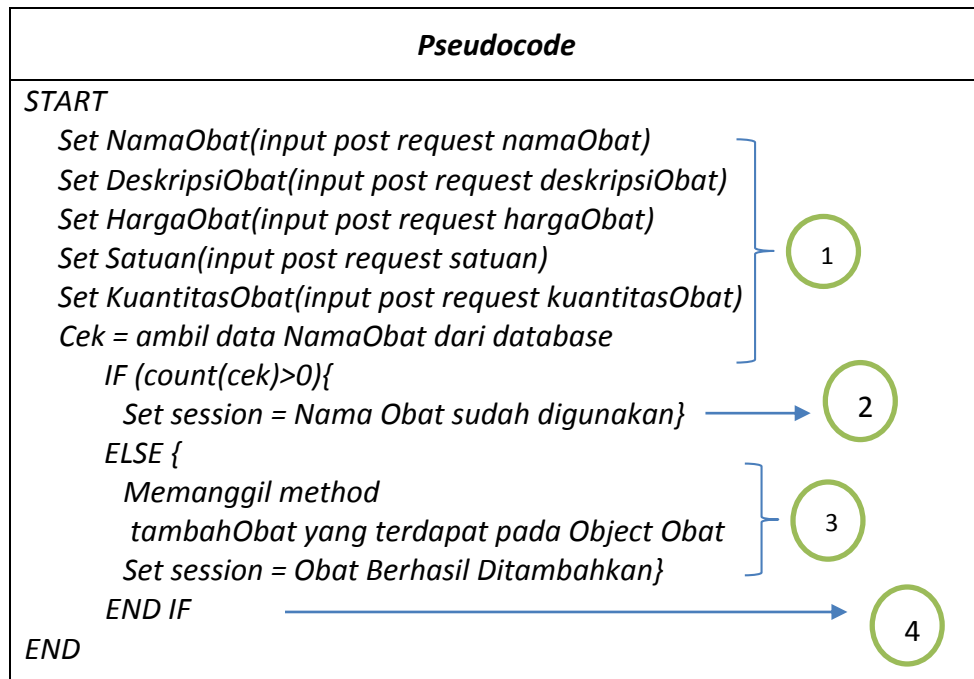
**Tabel 6.5 Hasil pengujian integrasi *method* tambahPengguna()**

No. Jalur	Prosedur Uji	Expected Result	Result	Status
1	Klas Driver memanggil <i>method</i> tambahPengguna() pada klas PenggunaController dengan set NamaPengguna("Alvina Ulfah Rusmayuni"), set Username("alvinaulfah"), set Password("doremifa"), set IdPekerjaan("2"), set StatusKerja("Aktif"), set Kehadiran("Tidak Hadir")	Semua data pengguna disimpan ke dalam <i>database</i> dan menampilkan pesan "Data berhasil disimpan"	Semua data Pengguna disimpan ke dalam <i>database</i> dan menampilkan pesan "Data berhasil disimpan"	Valid
2	Klas Driver memanggil <i>method</i> tambahPengguna() pada klas PenggunaController dengan set NamaPengguna("Alvina Ulfah Rusmayuni"), set Username("alvinaulfah"), set Password("doremifa"), set IdPekerjaan("2"), set StatusKerja("Aktif"), set Kehadiran("Tidak Hadir")	Semua data pengguna gagal disimpan ke dalam <i>database</i> dan menampilkan pesan "Username Sudah Digunakan"	Semua data pengguna gagal disimpan ke dalam <i>database</i> dan menampilkan pesan "Username Sudah Digunakan"	Valid



### 6.2.3 Pengujian Integrasi *Method* tambahObat() Klas ObatController

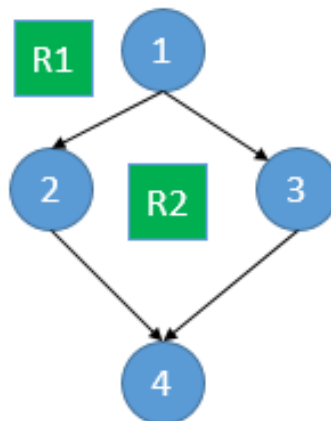
#### 1. Pseudocode



#### 2. Basis Path Testing

##### a. Flow Graph

Gambar 6.6 berikut ini adalah *flow graph method* tambahObat() dari klas ObatController.



**Gambar 6.6 Flow graph pengujian integrasi *method* tambahObat() klas ObatController**

##### b. Cyclomatic Complexity (V(G))

1.  $V(G) = \text{jumlah region} = 2$
2.  $V(G) = (\text{jumlah edge} - \text{jumlah node}) + 2 = (4 - 4) + 2 = 2$
3.  $V(G) = \text{jumlah predicate node} + 1 = 1 + 1 = 2$

c. *Independent Path*

- Jalur 1 : 1-2-4
- Jalur 2 : 1-3-4

Tabel 6.6 berikut merupakan kasus uji dan hasil pengujian *method* tambahObat klas ObatController.

**Tabel 6.6 Hasil pengujian integrasi *method* tambahObat()**

No. Jalur	Prosedur Uji	Expected Result	Result	Status
1	Klas Driver memanggil <i>method</i> tambahObat() pada klas ObatController dengan set NamaObat("Allopurinol 100mg"), set DeskripsiObat("Obat Asam Urat"), set HargaObat("2500"), set Satuan("Keping"), set KuantitasObat("20")	Semua data obat disimpan ke dalam <i>database</i> dan menampilkan pesan "Data berhasil disimpan"	Semua data obat disimpan ke dalam <i>database</i> dan menampilkan pesan "Data berhasil disimpan"	Valid
2	Klas Driver memanggil <i>method</i> tambahObat() pada klas ObatController dengan set NamaObat("Allopurinol 100mg"), set DeskripsiObat("Obat Asam Urat"), set HargaObat("2500"), set Satuan("Keping"), set KuantitasObat("20")	Semua data obat gagal disimpan ke dalam <i>database</i> dan menampilkan pesan "Nama Obat Sudah Digunakan"	Semua data obat gagal disimpan ke dalam <i>database</i> dan menampilkan pesan "Nama Obat Sudah Digunakan"	Valid

## 6.3 Pengujian Validasi

Pengujian validasi menguji 20 kebutuhan fungsional yang terdapat pada Sistem Pengelolaan Rekam Medis Pasien Berbasis Web.

### 6.3.1 Pengujian Validasi *Login*

#### 6.3.1.1 Pengujian Validasi *Login*

Tabel 6.7 merupakan tabel pengujian validasi untuk kasus uji *login*.

**Tabel 6.7 Pengujian validasi *Login***

Kode Kebutuhan	SIPREM-F-01
Nama Kasus Uji	<i>login</i>
Langkah-langkah	1. Membuka halaman <i>login</i> SIPREM 2. Memasukkan <i>username</i> = helmant dan <i>password</i> = samarinda 3. Menekan tombol <i>login</i>
Hasil yang diharapkan	Aktor akan teridentifikasi sebagai petugas administrasi dan masuk ke dalam sistem
Hasil	Aktor teridentifikasi sebagai petugas administrasi dan masuk ke dalam sistem
Status	valid

#### 6.3.1.2 Pengujian Validasi *Login* Alternatif Satu

Tabel 6.8 merupakan tabel pengujian validasi untuk kasus uji *login* alternatif flow 1a.

**Tabel 6.8 Pengujian validasi *Login* alternative satu**

Kode Kebutuhan	SIPREM-F-01
Nama Kasus Uji	<i>Login</i> alternative satu
Langkah-langkah	1. Membuka halaman <i>login</i> SIPREM 2. Memasukkan <i>username</i> = 123tes123 3. Menekan tombol <i>login</i>
Hasil yang diharapkan	Sistem menampilkan pesan untuk segera mengisi kolom yang kosong
Hasil	Sistem menampilkan pesan untuk segera mengisi kolom yang kosong
Status	valid

#### 6.3.1.3 Pengujian Validasi *Login* Alternatif Dua

Tabel 6.9 merupakan tabel pengujian validasi untuk kasus uji *login* alternatif flow 2a.

**Tabel 6.9 Pengujian validasi *Login* alternatif dua**

Kode Kebutuhan	SIPREM-F-01
Nama Kasus Uji	<i>Login</i> alternative dua
Langkah-langkah	1. Membuka halaman <i>login</i> SIPREM 2. Memasukkan <i>username</i> = helmante dan <i>password</i> = samarinda 3. Menekan tombol <i>login</i>
Hasil yang diharapkan	Sistem menampilkan pesan bahwa <i>username</i> salah
Hasil	Sistem menampilkan pesan bahwa <i>username</i> salah
Status	valid

#### 6.3.1.4 Pengujian Validasi *Login* Alternatif Tiga

Tabel 6.10 merupakan tabel pengujian validasi untuk kasus uji *login* alternatif flow 2b.

**Tabel 6.10 Pengujian validasi *Login* alternatif tiga**

Kode Kebutuhan	SIPREM-F-01
Nama Kasus Uji	<i>Login</i> alternative tiga
Langkah-langkah	1. Membuka halaman <i>login</i> SIPREM 2. Memasukkan <i>username</i> = helmant dan <i>password</i> = balikpapan 3. Menekan tombol <i>login</i>
Hasil yang diharapkan	Sistem menampilkan pesan bahwa <i>password</i> salah
Hasil	Sistem menampilkan pesan bahwa <i>password</i> salah
Status	valid

### 6.3.2 Pengujian Validasi *Logout*

Tabel 6.11 merupakan tabel pengujian validasi untuk kasus uji *logout*.

**Tabel 6.11 Pengujian validasi *Logout***

Kode Kebutuhan	SIPREM-F-02
Nama Kasus Uji	<i>logout</i>
Langkah-langkah	1. Menekan tombol <i>logout</i> pada <i>navigation bar</i> 2. Menekan tombol <i>yes</i>
Hasil yang diharapkan	Aktor akan keluar dari sistem, <i>session</i> aktor dihapus, dan menampilkan halaman <i>login</i>
Hasil	Aktor keluar dari sistem, <i>session</i> aktor dihapus, dan menampilkan halaman <i>login</i>
Status	valid

#### 6.3.2.1 Pengujian Validasi *Logout* Alternatif satu

Tabel 6.12 merupakan tabel pengujian validasi untuk kasus uji *logout* alternatif flow 3a.

**Tabel 6.12 Pengujian validasi *Logout* alternatif satu**

Kode Kebutuhan	SIPREM-F-02
Nama Kasus Uji	<i>Logout</i> alternatif satu
Langkah-langkah	1. Menekan tombol <i>logout</i> pada <i>navigation bar</i> 2. Menekan tombol <i>cancel</i>
Hasil yang diharapkan	Aktor tetap berada di dalam sistem
Hasil	Aktor tetap berada di dalam sistem
Status	valid

### 6.3.3 Pengujian Validasi Menambah Pengguna

#### 6.3.3.1 Pengujian Validasi Menambah Pengguna

Tabel 6.13 merupakan tabel pengujian validasi untuk kasus uji menambah pengguna.

**Tabel 6.13 Pengujian validasi Menambah Pengguna**

Kode Kebutuhan	SIPREM-F-03
Nama Kasus Uji	menambah pengguna
Langkah-langkah	1. Membuka halaman untuk menambah pengguna 2. Memasukkan nama pengguna = Alvina Ulfah, <i>username</i> = alvinaulfah, <i>password</i> = doremifa, dan memilih pekerjaan = Dokter 3. Menekan tombol <i>submit</i>
Hasil yang diharapkan	Data pengguna baru akan tersimpan pada <i>database</i> dan pengguna baru dapat menggunakan sistem sesuai jenis pekerjaannya
Hasil	Data pengguna baru tersimpan pada <i>database</i> dan menampilkan pesan bahwa pengguna berhasil ditambahkan
Status	valid

#### 6.3.3.2 Pengujian Validasi Menambah Pengguna Alternatif Satu

Tabel 6.14 merupakan tabel pengujian validasi untuk kasus uji menambah pengguna alternatif flow 3a.

**Tabel 6.14 Pengujian validasi Menambah Pengguna Alternative satu**

Kode Kebutuhan	SIPREM-F-03
Nama Kasus Uji	menambah pengguna alternative satu
Langkah-langkah	1. Membuka halaman untuk menambah pengguna 2. Memasukkan <i>username</i> = alvinaulfah, <i>password</i> = doremifa, memilih pekerjaan = Dokter dan tidak memasukkan nama pengguna 3. Menekan tombol <i>submit</i>

Hasil yang diharapkan	Sistem akan menampilkan pemberitahuan bahwa terdapat kolom kosong yang harus segera diisi
Hasil	Sistem menampilkan pemberitahuan bahwa terdapat kolom kosong yang harus segera diisi
Status	valid

### 6.3.3.3 Pengujian validasi Menambah Pengguna alternatif dua

Tabel 6.15 merupakan tabel pengujian validasi untuk kasus uji menambah pengguna alternatif flow 3b.

**Tabel 6.15 Pengujian validasi Menambah Pengguna alternatif dua**

Kode Kebutuhan	SIPREM-F-03
Nama Kasus Uji	menambah pengguna alternative dua
Langkah-langkah	<ol style="list-style-type: none"> <li>1. Membuka halaman untuk menambah pengguna</li> <li>2. Memasukkan <i>username</i> = alvinaulfah, <i>password</i> = doremifa, memilih pekerjaan = Dokter dan nama pengguna = Alvina Ulfah</li> <li>3. Menekan tombol <i>submit</i></li> </ol>
Hasil yang diharapkan	Sistem akan menampilkan pemberitahuan bahwa <i>username</i> telah digunakan dan kembali ke halaman tambah pengguna
Hasil	Sistem menampilkan pemberitahuan bahwa <i>username</i> telah digunakan dan kembali ke halaman tambah pengguna
Status	valid

### 6.3.4 Pengujian Validasi Melihat Daftar Pengguna

Tabel 6.16 merupakan tabel pengujian validasi untuk kasus uji melihat daftar pengguna.

**Tabel 6.16 Pengujian validasi Melihat Daftar Pengguna**

Kode Kebutuhan	SIPREM-F-04
----------------	-------------

Nama Kasus Uji	melihat daftar pengguna
Langkah-langkah	1. Menekan tombol lihat pengguna
Hasil yang diharapkan	Sistem akan menampilkan daftar pengguna sistem yang terdapat pada <i>database</i>
Hasil	Sistem menampilkan daftar pengguna sistem yang terdapat pada <i>database</i>
Status	valid

### 6.3.5 Pengujian Validasi Mengubah Data Pengguna

#### 6.3.5.1 Pengujian Validasi Mengubah Data Pengguna

Tabel 6.17 merupakan tabel pengujian validasi untuk kasus uji mengubah data pengguna.

**Tabel 6.17 Pengujian validasi Mengubah Data Pengguna**

Kode Kebutuhan	SIPREM-F-05
Nama Kasus Uji	mengubah data pengguna
Langkah-langkah	<ol style="list-style-type: none"> <li>1. Menekan tombol ubah data pengguna pada data pengguna bernama Alvina Ulfah</li> <li>2. Sistem menampilkan halaman ubah data pengguna</li> <li>3. Mengubah nama pengguna = Alvina Ulfah Rusmayuni</li> </ol>
Hasil yang diharapkan	Sistem akan menyimpan perubahan data dan menampilkan daftar pengguna sistem yang terdapat pada <i>database</i>
Hasil	Sistem menyimpan perubahan data dan menampilkan daftar pengguna sistem yang terdapat pada <i>database</i>
Status	valid



#### 6.3.5.2 Pengujian validasi mengubah data pengguna alternatif satu

Tabel 6.18 merupakan tabel pengujian validasi untuk kasus uji mengubah data pengguna alternatif flow 3a.

**Tabel 6.18 Pengujian validasi Mengubah Data Pengguna alternatif flow satu**

Kode Kebutuhan	SIPREM-F-05
Nama Kasus Uji	Mengubah data pengguna alternatif satu
Langkah-langkah	1. Membuka halaman untuk menambah pengguna 2. Menekan tombol <i>cancel</i>
Hasil yang diharapkan	Proses perubahan data pengguna dibatalkan dan kembali ke halaman daftar pengguna
Hasil	Proses perubahan data pengguna dibatalkan dan kembali ke halaman daftar pengguna
Status	valid

#### 6.3.5.3 Pengujian validasi mengubah data pengguna alternatif dua

Tabel 6.19 merupakan tabel pengujian validasi untuk kasus uji mengubah data pengguna alternatif flow 4a.

**Tabel 6.19 Pengujian validasi Mengubah Data Pengguna alternatif dua**

Kode Kebutuhan	SIPREM-F-05
Nama Kasus Uji	Mengubah data pengguna alternatif dua
Langkah-langkah	1. Membuka halaman untuk menambah pengguna 2. Menekan tombol <i>submit</i>
Hasil yang diharapkan	Sistem menampilkan pesan untuk mengisi kolom yang kosong
Hasil	Sistem menampilkan pesan untuk mengisi kolom yang kosong
Status	valid

#### 6.3.5.4 Pengujian Validasi Mengubah Data Pengguna Alternatif Tiga

Tabel 6.20 merupakan tabel pengujian validasi untuk kasus uji mengubah data pengguna alternatif flow 4b.

**Tabel 6.20 Pengujian validasi Mengubah Data Pengguna alternative satu**

Kode Kebutuhan	SIPREM-F-05
Nama Kasus Uji	mengubah data pengguna alternative satu
Langkah-langkah	1. Menekan tombol ubah data pengguna pada data pengguna bernama Alvina Ulfah 2. Memasukkan nama pengguna = Alvina Ulfah, <i>username</i> = sodiqdiq, <i>password</i> = doremifa, dan memilih pekerjaan = Dokter
Hasil yang diharapkan	Sistem akan menampilkan pemberitahuan bahwa <i>username</i> sudah digunakan dan data pengguna tidak berhasil dirubah
Hasil	Sistem menampilkan pemberitahuan bahwa <i>username</i> sudah digunakan dan data pengguna tidak berhasil dirubah
Status	valid

### **6.3.6 Pengujian Validasi Melihat Daftar Pasien Khusus Petugas Administrasi**

Tabel 6.21 merupakan tabel pengujian validasi untuk kasus uji melihat daftar pasien khusus petugas administrasi.

**Tabel 6.21 Pengujian validasi Melihat Daftar Pasien Khusus Petugas Administrasi**

Kode Kebutuhan	SIPREM-F-06
Nama Kasus Uji	melihat daftar pasien khusus petugas administrasi
Langkah-langkah	1. Menekan tombol lihat daftar pasien
Hasil yang diharapkan	Sistem akan menampilkan daftar pasien khusus petugas administrasi yang terdapat pada <i>database</i>
Hasil	Sistem berhasil menampilkan daftar pasien khusus petugas administrasi yang terdapat pada <i>database</i>
Status	valid

### 6.3.7 Pengujian Validasi Melihat Daftar Pasien Khusus Dokter

Tabel 6.22 merupakan tabel pengujian validasi untuk kasus uji melihat daftar pasien khusus dokter.

**Tabel 6.22 Pengujian validasi Melihat Daftar Pasien Khusus Dokter**

Kode Kebutuhan	SIPREM-F-07
Nama Kasus Uji	melihat daftar pasien khusus dokter
Langkah-langkah	1. Menekan tombol lihat daftar pasien
Hasil yang diharapkan	Sistem akan menampilkan daftar pasien khusus dokter yang terdapat pada <i>database</i>
Hasil	Sistem berhasil menampilkan daftar pasien khusus dokter yang terdapat pada <i>database</i>
Status	valid

### 6.3.8 Pengujian Validasi Menambah Pasien Baru

#### 6.3.8.1 Pengujian Validasi Menambah Pasien Baru

Tabel 6.19 merupakan tabel pengujian validasi untuk kasus uji menambah pasien baru yang berhasil.

**Tabel 6.23 Pengujian validasi Menambah Pasien Baru**

Kode Kebutuhan	SIPREM-F-08
Nama Kasus Uji	menambah pasien baru
Langkah-langkah	<ol style="list-style-type: none"><li>1. Menekan tombol tambah pasien</li><li>2. Sistem menampilkan halaman untuk menambah pasien</li><li>3. Memasukkan nama pasien = Aslam Bryantama, nomor BPJS = 0000432143214, jenis kelamin = L, tempat lahir = Samarinda, tanggal lahir = 20 April 1996, Alamat = jalan proklamasi 3 nomor 12 RT. 18 nomor telepon = 081143782916</li><li>4. Menekan tombol <i>submit</i></li></ol>
Hasil yang diharapkan	Sistem akan menyimpan data pasien baru, menampilkan daftar pasien khusus petugas administrasi yang terdapat pada <i>database</i>

	dan menampilkan pesan bahwa pasien berhasil ditambahkan
Hasil	Sistem berhasil menyimpan data pasien baru, menampilkan daftar pasien khusus petugas administrasi yang terdapat pada <i>database</i> dan menampilkan pesan bahwa pasien berhasil ditambahkan
Status	valid

#### 6.3.8.2 Pengujian Validasi Menambah Pasien Baru Alternatif Satu

Tabel 6.24 merupakan tabel pengujian validasi untuk kasus uji menambah pasien baru alternatif flow 3a.

**Tabel 6.24 Pengujian validasi Menambah Pasien Baru alternative satu**

Kode Kebutuhan	SIPREM-F-08
Nama Kasus Uji	menambah pasien baru alternative satu
Langkah-langkah	<ol style="list-style-type: none"> <li>1. Menekan tombol tambah pasien</li> <li>2. Sistem menampilkan halaman untuk menambah pasien</li> <li>3. Memasukkan nama pasien = NULL, nomor BPJS = -, jenis kelamin = L, tempat lahir = Samarinda, tanggal lahir = 20 April 1996, Alamat = jalan proklamasi 3 nomor 12 RT. 18 , nomor telepon = 081143782916</li> <li>4. Menekan tombol <i>submit</i></li> </ol>
Hasil yang diharapkan	Sistem tidak akan menyimpan data pasien baru dan akan menampilkan peringatan untuk mengisi kolom yang kosong
Hasil	Sistem tidak menyimpan data pasien baru dan menampilkan peringatan untuk mengisi kolom yang kosong
Status	valid

#### 6.3.8.3 Pengujian Validasi Menambah Pasien Baru Alternatif Dua

Tabel 6.25 merupakan tabel pengujian validasi untuk kasus uji menambah pasien baru alternatif flow 3b.

**Tabel 6.25 Pengujian validasi Menambah Pasien Baru alternatif dua**

Kode Kebutuhan	SIPREM-F-08
Nama Kasus Uji	menambah pasien baru alternatif dua
Langkah-langkah	<ol style="list-style-type: none"> <li>1. Menekan tombol tambah pasien</li> <li>2. Sistem menampilkan halaman untuk menambah pasien</li> <li>3. Memasukkan nama pasien = Aslam Bryantama, nomor BPJS = 0000432143214, jenis kelamin = L, tempat lahir = Samarinda, tanggal lahir = 20 April 1996, Alamat = jalan proklamasi 3 nomor 12 RT. 18 nomor telepon = 081143782916</li> <li>4. Menekan tombol <i>submit</i></li> </ol>
Hasil yang diharapkan	Sistem tidak akan menyimpan data pasien baru dan menampilkan pesan bahwa nomor BPJS sudah digunakan
Hasil	Sistem tidak menyimpan data pasien baru dan menampilkan pesan bahwa nomor BPJS sudah digunakan
Status	valid

#### 6.3.8.4 Pengujian validasi menambah pasien baru alternatif tiga

Tabel 6.26 merupakan tabel pengujian validasi untuk kasus uji menambah pasien baru alternatif flow 3c.

**Tabel 6.26 Pengujian validasi Menambah Pasien Baru alternatif tiga**

Kode Kebutuhan	SIPREM-F-08
Nama Kasus Uji	menambah pasien baru alternatif tiga
Langkah-langkah	<ol style="list-style-type: none"> <li>1. Menekan tombol tambah pasien</li> <li>2. Sistem menampilkan halaman untuk menambah pasien</li> <li>3. Menekan tombol <i>cancel</i></li> </ol>
Hasil yang diharapkan	Sistem menampilkan halaman utama
Hasil	Sistem menampilkan halaman utama
Status	valid

### 6.3.9 Pengujian Validasi Mengubah Data Pasien

#### 6.3.9.1 Pengujian Validasi Mengubah Data Pasien

Tabel 6.27 merupakan tabel pengujian validasi untuk kasus uji mengubah data pasien.

**Tabel 6.27 Pengujian validasi Mengubah Data Pasien**

Kode Kebutuhan	SIPREM-F-09
Nama Kasus Uji	mengubah data pasien
Langkah-langkah	<ol style="list-style-type: none"><li>1. Menekan tombol ubah data pasien pada pasien bernama Ragil Setyo</li><li>2. Sistem menampilkan halaman ubah data pengguna</li><li>3. Mengubah nama pasien = Ragil Setyo Utomo</li><li>4. Menekan tombol <i>submit</i></li></ol>
Hasil yang diharapkan	Sistem akan menyimpan perubahan data, menampilkan daftar pasien yang terdapat pada <i>database</i> dan menampilkan pesan bahwa data pasien berhasil dirubah
Hasil	Sistem menyimpan perubahan data, menampilkan daftar pasien yang terdapat pada <i>database</i> dan menampilkan pesan bahwa data pasien berhasil dirubah
Status	valid

#### 6.3.9.2 Pengujian validasi mengubah data pasien alternatif satu

Tabel 6.28 merupakan tabel pengujian validasi untuk kasus uji mengubah data pasien alternatif flow 3a.

**Tabel 6.28 Pengujian validasi Mengubah Data Pasien alternatif satu**

Kode Kebutuhan	SIPREM-F-09
Nama Kasus Uji	mengubah data pasien alternative satu
Langkah-langkah	<ol style="list-style-type: none"><li>1. Menekan tombol ubah data pasien pada pasien bernama Ragil Setyo</li><li>2. Menekan tombol <i>submit</i></li></ol>
Hasil yang diharapkan	Sistem menampilkan pesan untuk mengisi kolom yang kosong

Hasil	Sistem menampilkan pesan untuk mengisi kolom yang kosong
Status	valid

#### 6.3.9.3 Pengujian Validasi Mengubah Data Pasien Alternatif Dua

Tabel 6.29 merupakan tabel pengujian validasi untuk kasus uji mengubah data pasien alternatif flow 3b.

**Tabel 6.29 Pengujian validasi Mengubah Data Pasien alternatif dua**

Kode Kebutuhan	SIPREM-F-09
Nama Kasus Uji	mengubah data pasien alternative dua
Langkah-langkah	3. Menekan tombol ubah data pasien pada pasien bernama Ragil Setyo 4. Mengubah nomor BPJS menjadi 0000123412341
Hasil yang diharapkan	Proses perubahan data pasien dibatalkan, menampilkan halaman daftar pasien dan menampilkan pesan bahwa nomor BPJS sudah digunakan
Hasil	Proses perubahan data pasien dibatalkan, menampilkan halaman daftar pasien dan menampilkan pesan bahwa nomor BPJS sudah digunakan
Status	valid

#### 6.3.9.4 Pengujian Validasi Mengubah Data Pasien Alternatif Tiga

Tabel 6.30 merupakan tabel pengujian validasi untuk kasus uji mengubah data pasien alternatif flow 3c.

**Tabel 6.30 Pengujian validasi Mengubah Data Pasien alternatif tiga**

Kode Kebutuhan	SIPREM-F-09
Nama Kasus Uji	mengubah data pasien alternative tiga
Langkah-langkah	1. Menekan tombol ubah data pasien pada pasien bernama Ragil Setyo 2. Menekan tombol <i>cancel</i> pada halaman mengubah data pasien
Hasil yang diharapkan	Proses perubahan data pasien dibatalkan

Hasil	Proses perubahan data pasien dibatalkan
Status	valid

### 6.3.10 Pengujian Validasi Melihat Rekam Medis

Tabel 6.31 merupakan tabel pengujian validasi untuk kasus uji melihat rekam medis.

**Tabel 6.31 Pengujian validasi Melihat Rekam Medis**

Kode Kebutuhan	SIPREM-F-10
Nama Kasus Uji	melihat rekam medis
Langkah-langkah	1. Menekan tombol lihat rekam medis pada pasien yang ingin dilihat rekam medisnya
Hasil yang diharapkan	Sistem akan menampilkan rekam medis pasien yang terdapat pada <i>database</i>
Hasil	Sistem berhasil menampilkan rekam medis pasien yang terdapat pada <i>database</i>
Status	valid

### 6.3.11 Pengujian Validasi Melihat Daftar Obat Khusus Dokter

Tabel 6.32 merupakan tabel pengujian validasi untuk kasus uji melihat daftar obat khusus dokter.

**Tabel 6.32 Pengujian validasi Melihat Daftar Obat Khusus Dokter**

Kode Kebutuhan	SIPREM-F-11
Nama Kasus Uji	melihat daftar obat khusus dokter
Langkah-langkah	1. Menekan tombol lihat obat pada <i>menu bar</i>
Hasil yang diharapkan	Sistem akan menampilkan daftar obat khusus dokter yang terdapat pada <i>database</i>
Hasil	Sistem berhasil menampilkan daftar obat khusus dokter yang terdapat pada <i>database</i>
Status	valid



### 6.3.12 Pengujian Validasi Melihat Daftar Obat Khusus Apoteker

Tabel 6.33 merupakan tabel pengujian validasi untuk kasus uji melihat daftar obat khusus apoteker.

**Tabel 6.33 Pengujian validasi Melihat Daftar Obat Khusus Apoteker**

Kode Kebutuhan	SIPREM-F-12
Nama Kasus Uji	melihat daftar obat khusus apoteker
Langkah-langkah	1. Menekan tombol lihat obat pada <i>menu bar</i>
Hasil yang diharapkan	Sistem akan menampilkan daftar obat khusus apoteker yang terdapat pada <i>database</i>
Hasil	Sistem berhasil menampilkan daftar obat khusus apoteker yang terdapat pada <i>database</i>
Status	valid

### 6.3.13 Pengujian Validasi Menambah Obat Baru

#### 6.3.13.1 Pengujian Validasi Menambah Obat Baru

Tabel 6.34 merupakan tabel pengujian validasi untuk kasus uji menambah obat baru.

**Tabel 6.34 Pengujian validasi Menambah Obat Baru**

Kode Kebutuhan	SIPREM-F-13
Nama Kasus Uji	menambah obat baru
Langkah-langkah	1. Menekan tombol tambah obat 2. Sistem menampilkan halaman untuk menambah obat 3. Memasukkan nama obat = Ranitidin 50mg, deskripsi obat = obat lambung, harga obat = Rp 6000, kuantitas obat = 20 4. Menekan tombol <i>submit</i>
Hasil yang diharapkan	Sistem menyimpan data obat baru, menampilkan halaman daftar obat dan menampilkan pesan bahwa obat berhasil ditambahkan
Hasil	Sistem berhasil menyimpan data obat baru, menampilkan halaman daftar obat

	dan menampilkan pesan bahwa obat berhasil ditambahkan
Status	valid

#### 6.3.13.2 Pengujian validasi menambah obat baru alternatif satu

Tabel 6.35 merupakan tabel pengujian validasi untuk kasus uji menambah obat baru alternatif flow 3a.

**Tabel 6.35 Pengujian validasi Menambah Obat Baru alternatif satu**

Kode Kebutuhan	SIPREM-F-13
Nama Kasus Uji	menambah obat baru alternative satu
Langkah-langkah	<ol style="list-style-type: none"> <li>1. Menekan tombol tambah obat</li> <li>2. Sistem menampilkan halaman untuk menambah obat</li> <li>3. Memasukkan nama obat = Ranitidin 100mg, deskripsi obat = obat lambung, harga obat = 6000, satuan = keeping, kuantitas obat tidak diisi</li> <li>4. Menekan tombol <i>submit</i></li> </ol>
Hasil yang diharapkan	Sistem akan menampilkan pesan untuk mengisi kolom yang kosong
Hasil	Sistem akan menampilkan pesan untuk mengisi kolom yang kosong
Status	valid

#### 6.3.13.3 Pengujian Validasi Menambah Obat Baru Alternatif dua

Tabel 6.36 merupakan tabel pengujian validasi untuk kasus uji menambah obat baru alternatif flow 3b.

**Tabel 6.36 Pengujian validasi Menambah Obat Baru alternative dua**

Kode Kebutuhan	SIPREM-F-13
Nama Kasus Uji	menambah obat baru alternative dua
Langkah-langkah	<ol style="list-style-type: none"> <li>1. Menekan tombol tambah obat</li> <li>2. Sistem menampilkan halaman untuk menambah obat</li> <li>3. Memasukkan nama obat = Ranitidin 100mg, deskripsi obat = obat lambung,</li> </ol>

	<p>harga obat = 6000, satuan = keeping, kuantitas obat = 20</p> <p>4. Menekan tombol <i>submit</i></p>
Hasil yang diharapkan	Sistem akan menampilkan halaman daftar obat, menampilkan peringatan “Nama Obat Sudah Digunakan”, dan penambahan obat dibatalkan
Hasil	Sistem menampilkan halaman daftar obat, menampilkan peringatan “Nama Obat Sudah Digunakan”, dan penambahan obat dibatalkan
Status	valid

#### 6.3.13.4 Pengujian validasi menambah obat baru alternatif tiga

Tabel 6.37 merupakan tabel pengujian validasi untuk kasus uji menambah obat baru alternatif flow 3c.

**Tabel 6.37 Pengujian validasi Menambah Obat Baru alternatif tiga**

Kode Kebutuhan	SIPREM-F-13
Nama Kasus Uji	menambah obat baru alternative tiga
Langkah-langkah	<ol style="list-style-type: none"> <li>1. Menekan tombol tambah obat</li> <li>2. Sistem menampilkan halaman untuk menambah obat</li> <li>3. Menekan tombol <i>cancel</i></li> </ol>
Hasil yang diharapkan	Sistem akan menampilkan halaman utama
Hasil	Sistem menampilkan halaman utama
Status	valid

#### 6.3.14 Pengujian Validasi Mengubah Data Obat

##### 6.3.14.1 Pengujian Validasi Mengubah Data Obat

Tabel 6.38 merupakan tabel pengujian validasi untuk kasus uji mengubah data obat.

**Tabel 6.38 Pengujian validasi Mengubah Data Obat**

Kode Kebutuhan	SIPREM-F-14
Nama Kasus Uji	mengubah data obat

Langkah-langkah	<ol style="list-style-type: none"> <li>1. Menekan tombol ubah data obat pada obat bernama Ranitidin 100mg</li> <li>2. Sistem menampilkan halaman untuk mengubah data obat</li> <li>3. Mengubah harga obat = 7000</li> <li>4. Menekan tombol <i>submit</i></li> </ol>
Hasil yang diharapkan	Sistem menyimpan perubahan data obat, menampilkan halaman daftar obat dan menampilkan pesan bahwa data obat berhasil dirubah
Hasil	Sistem menyimpan perubahan data obat, menampilkan halaman daftar obat dan menampilkan pesan bahwa data obat berhasil dirubah
Status	valid

#### 6.3.14.2 Pengujian Validasi Mengubah Data Obat Alternatif Satu

Tabel 6.39 merupakan tabel pengujian validasi untuk kasus uji mengubah data obat alternatif flow 3a.

**Tabel 6.39 Pengujian validasi Mengubah Data Obat alternatif satu**

Kode Kebutuhan	SIPREM-F-14
Nama Kasus Uji	mengubah data obat alternative satu
Langkah-langkah	<ol style="list-style-type: none"> <li>1. Menekan tombol ubah data obat pada obat bernama Ranitidin 100mg</li> <li>2. Sistem menampilkan halaman untuk mengubah data obat</li> <li>3. Mengubah nama obat = Allopurinol 100 mg</li> <li>4. Menekan tombol <i>submit</i></li> </ol>
Hasil yang diharapkan	Sistem menampilkan halaman daftar obat, menampilkan peringatan “Nama Obat Sudah Digunakan”, dan perubahan data obat dibatalkan
Hasil	Sistem menampilkan halaman daftar obat, menampilkan peringatan “Nama Obat Sudah Digunakan”, dan perubahan data obat dibatalkan

Status	valid
--------	-------

#### 6.3.14.3 Pengujian validasi mengubah data obat alternatif dua

Tabel 6.40 merupakan tabel pengujian validasi untuk kasus uji mengubah data obat alternatif flow 3b.

**Tabel 6.40 Pengujian validasi Mengubah Data Obat alternatif dua**

Kode Kebutuhan	SIPREM-F-14
Nama Kasus Uji	mengubah data obat alternative dua
Langkah-langkah	<ol style="list-style-type: none"> <li>1. Menekan tombol ubah data obat pada obat bernama Ranitidin 100mg</li> <li>2. Sistem menampilkan halaman untuk mengubah data obat</li> <li>3. Menekan tombol <i>cancel</i></li> </ol>
Hasil yang diharapkan	Sistem menampilkan halaman daftar obat
Hasil	Sistem menampilkan halaman daftar obat
Status	valid

#### 6.3.15 Pengujian Validasi Menambah Hasil Pemeriksaan

##### 6.3.15.1 Pengujian Validasi Menambah Hasil Pemeriksaan

Tabel 6.41 merupakan tabel pengujian validasi untuk kasus uji menambah hasil pemeriksaan.

**Tabel 6.41 Pengujian validasi Menambah Hasil Pemeriksaan**

Kode Kebutuhan	SIPREM-F-15
Nama Kasus Uji	menambah hasil pemeriksaan
Langkah-langkah	<ol style="list-style-type: none"> <li>1. Menekan tombol Lihat Antrian</li> <li>2. Sistem menampilkan halaman untuk menambah hasil pemeriksaan</li> <li>3. Mengisi anamnesa = “demam, sakit kepala, lelah, berkeringat, dan sakit perut”, pemeriksaan fisik = “perut sakit, lidah merah”, diagnosis = tipes, terapi = obat amoxan drop 1 botol, catatan = amoxan 2x3</li> <li>4. Menekan tombol <i>submit</i></li> </ol>

Hasil yang diharapkan	Sistem menyimpan data medis pasien ke dalam <i>database</i> , menampilkan halaman utama, dan menampilkan pesan bahwa data medis berhasil ditambahkan
Hasil	Sistem menyimpan data medis pasien ke dalam <i>database</i> , menampilkan halaman utama, dan menampilkan pesan bahwa data medis berhasil ditambahkan
Status	valid

#### 6.3.15.2 Pengujian Validasi Menambah Hasil Pemeriksaan Alternatif satu

Tabel 6.42 merupakan tabel pengujian validasi untuk kasus uji menambah hasil pemeriksaan alternatif flow 3a.

**Tabel 6.42 Pengujian validasi Menambah Hasil Pemeriksaan alternatif dua**

Kode Kebutuhan	SIPREM-F-15
Nama Kasus Uji	menambah hasil pemeriksaan alternative satu
Langkah-langkah	<ol style="list-style-type: none"> <li>1. Menekan tombol lihat antrian</li> <li>2. Sistem menampilkan halaman untuk menambah hasil pemeriksaan</li> <li>3. Menekan tombol <i>cancel</i></li> </ol>
Hasil yang diharapkan	Sistem akan menampilkan halaman utama
Hasil	Sistem menampilkan utama
Status	valid

#### 6.3.16 Pengujian Validasi Mencetak Rekam Medis

Tabel 6.43 merupakan tabel pengujian validasi untuk kasus uji mencetak rekam medis yang berhasil.

**Tabel 6.43 Pengujian validasi Mencetak Rekam Medis**

Kode Kebutuhan	SIPREM-F-16
Nama Kasus Uji	mencetak rekam medis
Langkah-langkah	<ol style="list-style-type: none"> <li>1. Menekan tombol cetak rekam medis pada pasien bernama Namira Syahda Mayagita</li> <li>2. Menekan tombol <i>ok</i> pada halaman cetak rekam medis</li> </ol>

Hasil yang diharapkan	Sistem akan menyimpan data rekam medis pasien dalam bentuk <i>pdf</i>
Hasil	Sistem menyimpan data rekam medis pasien dalam bentuk <i>pdf</i>
Status	valid

### 6.3.17 Pengujian Validasi Melihat Antrian Pasien Khusus Petugas Administrasi

Tabel 6.44 merupakan tabel pengujian validasi untuk kasus uji melihat antrian pasien khusus petugas administrasi.

**Tabel 6.44 Pengujian validasi Melihat Antrian Pasien Khusus Petugas Administrasi**

Kode Kebutuhan	SIPREM-F-17
Nama Kasus Uji	melihat antrian pasien khusus petugas administrasi
Langkah-langkah	1. Memilih menu lihat antrian
Hasil yang diharapkan	Sistem akan menampilkan halaman antrian pasien khusus petugas administrasi
Hasil	Sistem menampilkan halaman antrian pasien khusus petugas administrasi
Status	valid

### 6.3.18 Pengujian Validasi Menambahkan Pasien Ke dalam Antrian

#### 6.3.18.1 Pengujian Validasi Menambahkan Pasien Ke dalam Antrian

Tabel 6.45 merupakan tabel pengujian validasi untuk kasus uji menambahkan pasien ke dalam antrian.

**Tabel 6.45 Pengujian validasi Menambahkan Pasien Ke dalam Antrian**

Kode Kebutuhan	SIPREM-F-18
Nama Kasus Uji	menambahkan pasien ke dalam antrian
Langkah-langkah	<ol style="list-style-type: none"> <li>1. Memilih menu lihat pasien</li> <li>2. Sistem menampilkan daftar pasien</li> <li>3. Menekan tombol masukkan ke dalam antrian pada pasien bernama Namira Syahda Mayagita</li> </ol>

Hasil yang diharapkan	Sistem memasukkan pasien ke dalam antrian sistem, menampilkan halaman antrian khusus petugas administrasi, dan menampilkan pesan bahwa pasien berhasil ditambahkan ke dalam antrian.
Hasil	Sistem memasukkan pasien ke dalam antrian sistem, menampilkan halaman antrian khusus petugas administrasi, dan menampilkan pesan bahwa pasien berhasil ditambahkan ke dalam antrian.
Status	valid

### 6.3.18.2 Pengujian Validasi Menambahkan Pasien Ke dalam Antrian Alternatif Satu

Tabel 6.46 merupakan tabel pengujian validasi untuk kasus uji menambahkan pasien ke dalam antrian alternatif flow 2a.

**Tabel 6.46 Pengujian validasi Menambahkan Pasien Ke dalam Antrian alternatif satu**

Kode Kebutuhan	SIPREM-F-18
Nama Kasus Uji	menambahkan pasien ke dalam antrian alternative satu
Langkah-langkah	<ol style="list-style-type: none"> <li>1. Memilih menu lihat pasien</li> <li>2. Sistem menampilkan daftar pasien</li> <li>3. Menekan tombol masukkan ke dalam antrian pada pasien bernama Namira Syahda Mayagita</li> </ol>
Hasil yang diharapkan	Sistem akan memberikan pesan bahwa pasien sudah mengantri dan nomor antrian belum terlewatkan
Hasil	Sistem memberikan pesan bahwa pasien sudah mengantri dan nomor antrian belum terlewatkan
Status	valid

### 6.3.19 Pengujian Validasi Membatalkan Antrian Pasien

#### 6.3.19.1 Pengujian validasi membatalkan antrian pasien

Tabel 6.47 merupakan tabel pengujian validasi untuk kasus uji membatalkan antrian pasien.



**Tabel 6.47 Pengujian validasi Membatalkan Antrian Pasien**

Kode Kebutuhan	SIPREM-F-19
Nama Kasus Uji	membatalkan antrian pasien
Langkah-langkah	<ol style="list-style-type: none"> <li>1. Memilih menu lihat antrian</li> <li>2. Sistem menampilkan daftar antrian</li> <li>3. Menekan tombol batalkan antrian pada pasien bernama Namira Syahda Mayagita</li> <li>4. Menekan tombol “ya” pada <i>pop-up</i></li> </ol>
Hasil yang diharapkan	Sistem akan membatalkan antrian pasien dan mengeluarkan pasien dari dalam antrian
Hasil	Sistem membatalkan pasien dan mengeluarkan pasien dari dalam antrian
Status	valid

#### 6.3.19.2 Pengujian validasi membatalkan antrian pasien alternatif satu

Tabel 6.48 merupakan tabel pengujian validasi untuk kasus uji membatalkan antrian pasien alternatif flow 3a.

**Tabel 6.48 Pengujian validasi Membatalkan Antrian Pasien alternatif satu**

Kode Kebutuhan	SIPREM-F-19
Nama Kasus Uji	membatalkan antrian pasien alternatif satu
Langkah-langkah	<ol style="list-style-type: none"> <li>1. Memilih menu lihat antrian</li> <li>2. Sistem menampilkan daftar antrian</li> <li>3. Menekan tombol batalkan antrian pada pasien bernama Namira Syahda Mayagita</li> <li>4. Menekan tombol “tidak” pada <i>“pop-up”</i></li> </ol>
Hasil yang diharapkan	Sistem akan tidak jadi membatalkan antrian pasien
Hasil	Sistem tidak jadi membatalkan antrian pasien
Status	valid

### 6.3.20 Pengujian Validasi Menambahkan Pasien Ke dalam Ruang Pemeriksaan

#### 6.3.20.1 Pengujian Validasi Menambahkan Pasien ke dalam Ruang Pemeriksaan

Tabel 6.49 merupakan tabel pengujian validasi untuk kasus uji menambahkan pasien ke dalam ruang pemeriksaan.

**Tabel 6.49 Pengujian validasi Menambahkan Pasien ke dalam Ruang Pemeriksaan**

Kode Kebutuhan	SIPREM-F-20
Nama Kasus Uji	menambahkan pasien ke dalam ruang pemeriksaan
Langkah-langkah	<ol style="list-style-type: none"><li>1. Memilih menu lihat antrian</li><li>2. Sistem menampilkan daftar antrian</li><li>3. Menekan tombol masukkan ke dalam ruangan pada pasien bernama Namira Syahda Mayagita</li><li>4. Sistem menampilkan <i>pop-up</i> yang berisi pilihan untuk memasukkan pasien ke dalam ruang pemeriksaan atau <i>cancel</i></li><li>5. Menekan tombol masukkan pasien ke dalam ruang pemeriksaan</li></ol>
Hasil yang diharapkan	Sistem akan menampilkan pesan bahwa pasien sukses dimasukkan ke dalam ruang pemeriksaan
Hasil	Sistem menampilkan pesan bahwa pasien sukses dimasukkan ke dalam ruang pemeriksaan
Status	valid

#### 6.3.20.2 Pengujian Validasi Menambahkan Pasien ke dalam Ruang Pemeriksaan Alternatif Satu

Tabel 6.40 merupakan tabel pengujian validasi untuk kasus uji menambahkan pasien ke dalam ruang pemeriksaan alternatif flow 3a.

**Tabel 6.50 Pengujian validasi Menambahkan Pasien ke dalam Ruang Pemeriksaan alternatif satu**

Kode Kebutuhan	SIPREM-F-20
----------------	-------------

Nama Kasus Uji	menambahkan pasien ke dalam ruang pemeriksaan alternative satu
Langkah-langkah	<ol style="list-style-type: none"> <li>1. Memilih menu lihat antrian</li> <li>2. Sistem menampilkan daftar antrian</li> <li>3. Menekan tombol masukkan ke dalam ruangan pada pasien bernama Namira Syahda Mayagita</li> <li>4. Sistem menampilkan <i>pop-up</i> yang berisi pilihan untuk memasukkan pasien ke dalam ruang pemeriksaan atau <i>cancel</i></li> <li>5. Menekan tombol <i>cancel</i></li> </ol>
Hasil yang diharapkan	Sistem akan menampilkan halaman antrian pasien dan pasien tidak dimasukkan ke dalam ruang pemeriksaan
Hasil	Sistem menampilkan halaman antrian pasien dan pasien tidak dimasukkan ke dalam ruang pemeriksaan
Status	valid

### 6.3.20.3 Pengujian Validasi Menambahkan Pasien ke dalam Ruang Pemeriksaan Alternatif Dua

Tabel 6.51 merupakan tabel pengujian validasi untuk kasus uji menambahkan pasien ke dalam ruang pemeriksaan alternatif 3b.

**Tabel 6.51 Pengujian validasi Menambahkan Pasien ke dalam Ruang Pemeriksaan alternatif dua**

Kode Kebutuhan	SIPREM-F-20
Nama Kasus Uji	menambahkan pasien ke dalam ruang pemeriksaan alternative dua
Langkah-langkah	<ol style="list-style-type: none"> <li>1. Memilih menu lihat antrian</li> <li>2. Sistem menampilkan daftar antrian</li> <li>3. Menekan tombol masukkan ke dalam ruangan pada pasien bernama Namira Syahda Mayagita</li> </ol>
Hasil yang diharapkan	Sistem akan menampilkan <i>pop-up</i> yang berisi tombol memasukkan pasien ke dalam ruang pemeriksaan namun ter-

	<i>disable</i> karena status ruangan sedang terdapat pasien lain dan tombol <i>cancel</i>
Hasil	Sistem menampilkan <i>pop-up</i> yang berisi tombol memasukkan pasien ke dalam ruang pemeriksaan namun ter- <i>disable</i> karena status ruangan sedang terdapat pasien lain dan tombol <i>cancel</i>
Status	valid

#### 6.3.20.4 Pengujian Validasi Menambahkan Pasien ke dalam Ruang Pemeriksaan Alternatif Tiga

Tabel 6.52 merupakan tabel pengujian validasi untuk kasus uji menambahkan pasien ke dalam ruang pemeriksaan alternatif flow 3c.

**Tabel 6.52 Pengujian validasi Menambahkan Pasien ke dalam Ruang Pemeriksaan alternatif tiga**

Kode Kebutuhan	SIPREM-F-20
Nama Kasus Uji	menambahkan pasien ke dalam ruang pemeriksaan alternative tiga
Langkah-langkah	<ol style="list-style-type: none"> <li>1. Memilih menu lihat antrian</li> <li>2. Sistem menampilkan daftar antrian</li> <li>3. Menekan tombol masukkan ke dalam ruangan pada pasien bernama Namira Syahda Mayagita</li> </ol>
Hasil yang diharapkan	Sistem akan menampilkan <i>pop-up</i> yang berisi tombol memasukkan pasien ke dalam ruang pemeriksaan namun ter- <i>disable</i> karena dokter belum masuk ke dalam sistem dan tombol <i>cancel</i>
Hasil	Sistem menampilkan <i>pop-up</i> yang berisi tombol memasukkan pasien ke dalam ruang pemeriksaan namun ter- <i>disable</i> karena dokter belum masuk ke dalam sistem dan tombol <i>cancel</i>
Status	valid

## BAB 7 PENUTUP

### 7.1 Kesimpulan

1. Berdasarkan tahap kebutuhan yang telah dilakukan, diperoleh:
  - a. Lima jenis aktor yang dapat menggunakan sistem pengelolaan rekam medis pasien berbasis web, yaitu pengguna, admin, petugas administrasi, dokter, dan apoteker.
  - b. 20 kebutuhan fungsional yang dimiliki oleh sistem pengelolaan rekam medis pasien berbasis web. Beberapa fungsionalitas utama sistem dari hasil analisis ialah sistem harus mampu menyediakan fungsi menambahkan hasil pemeriksaan, sistem harus mampu menyediakan fungsi menampilkan rekam medis pasien, dan sistem harus mampu menyediakan fungsi mencetak rekam medis pasien.
  - c. *Use case diagram* yang terdiri dari 20 *use case*. Aktor Pengguna memiliki satu *use case*, aktor Dokter yang merupakan spesialisasi dari Pengguna memiliki tujuh *use case*, aktor Apoteker yang merupakan spesialisasi dari Pengguna memiliki empat *use case*, aktor Petugas Administrasi yang merupakan spesialisasi dari Pengguna memiliki delapan *use case*, dan aktor Admin yang merupakan spesialisasi dari Pengguna memiliki empat *use case*.
  - d. 20 *use case scenario* yang menjelaskan alur dari *use case* yang telah dibuat.
2. Berdasarkan tahap perancangan dan Implementasi yang telah dilakukan pada “Sistem Pengelolaan Rekam Medis Pasien Berbasis Web. Studi Kasus: Klinik Dr. Syarif”, diperoleh:
  - a. Proses perancangan menghasilkan *sequence diagram*, *class diagram*, perancangan data, perancangan komponen, dan perancangan antarmuka. Beberapa entitas utama dari hasil perancangan ialah klas Pasien, klas Obat, dan klas DataMedis.
  - b. Proses implementasi menghasilkan spesifikasi sistem, batasan implementasi, implementasi data, implementasi kode program, dan implementasi antarmuka.
3. Berdasarkan pengujian “Sistem Pengelolaan Rekam Medis Pasien Berbasis Web. Studi Kasus: Klinik Dr. Syarif”, pengujian unit, pengujian integrasi, dan pengujian validasi yang dilakukan menghasilkan nilai valid sebesar 100%. Dapat disimpulkan, sistem layak untuk digunakan.

## **7.2 Saran**

Saran yang dapat diberikan untuk pengembangan “Sistem Pengelolaan Rekam Medis Pasien Berbasis Web. Studi Kasus: Klinik Dr. Syarif” yaitu:

1. Sistem dapat ditambahkan fitur transaksi dalam melakukan pembayaran obat, sehingga dapat membantu klinik untuk mengelola data pembayaran obat pasien.

## DAFTAR REFERENSI

- Anhar, 2010. *PHP & MySql Secara Otodidak*. Jakarta: PT. TransMedia.
- Ayuliana, et al. 2014. *Perancangan Sistem Informasi Manajemen Pada Klinik Skala Kecil*. S1. Universitas Binus. Tersedia di: Journal Binus <http://journal.binus.ac.id> [Diakses 10 Januari 2019].
- Booch, G., Jacobson, I., & Rumbaugh, J., 2007. *Object-Oriented Analysis and Design with Applications*. 3rd Edition. Boston: Addison-Wesley.
- Fowler, M., 2003. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. 3rd Edition. Boston: Addison Wesley.
- Hakim, L., 2010. *Membangun Web Berbasis PHP dengan Framework Codeigniter*. Yogyakarta: Lokomedia.
- Huda, M., 2010. *Membuat Aplikasi Database dengan Java, MySQL dan Netbeans*. Jakarta: Elex Media Komputindo.
- Huffman, E. K., 1994. *Health Information Management*. Brewyn: Physicians Record Company.
- Kumar, N., 2013. *Evolving a New Software Development Life Cycle Model SDLC-2013 with Client Satisfaction*. International Journal of Soft Computing and Engineering (IJSCE).
- Marsic, I., 2012. *Software Engineering*. New Jersey: Rutgers University.
- Peranginangin, K., 2006. *Aplikasi WEB dengan PHP dan MySQL*. Yogyakarta: Andi.
- Peraturan Menteri Kesehatan Republik Indonesia nomor 269 tahun 2008 Tentang Rekam Medis. Jakarta: Kementrian Kesehatan Republik Indonesia.
- Pressman, R. S., 2010. *Software Engineering : A Practitioner's Approach*. 7th Edition. New York: McGraw-Hill.
- Robbins, J. N., 2018. *Learning Web Design : A beginner's Guide To HTML, CSS, Javascript, and Web Graphics*. 5th Edition. Canada: O'Rilley Media, Inc.
- Rumbaugh, J., Jacobson, I., & Booch, G., 2004. *The Unified Modeling Language User Guide*. 2nd Edition. Germany: Addison-Wesley.
- Satzinger, J. W., et al. 2010. *System Analysts And Design in A Changing World*. 7th Edition. Boston: Cengage Learning.
- Sommerville, I., 2011. *Software Engineering*. 9th Edition. Hagerstown: Pearson.

Sparx Systems., 1996. *Enterprise Architect User Guide*. [Online] Tersedia di : [https://sparxsystems.com/enterprise\\_architect\\_user\\_guide/12.1/building\\_models/whatisuml.html](https://sparxsystems.com/enterprise_architect_user_guide/12.1/building_models/whatisuml.html) [Diakses 08 Juli 2019]

Widya, J. S. K., dan Migunani, 2014. *Rancang Bangun Sistem Informasi Rekam Medis Berbasis Web*. S1. STMIK ProVisi Semarang. Tersedia di: Portalgaruda <http://id.portalgaruda.org> [Diakses 10 Januari 2019].



## **LAMPIRAN A HASIL WAWANCARA**

Narasumber: Dr. Syarif Ismail

### **1. Bagaimana proses pengelolaan rekam medis yang selama ini dilakukan?**

Pertama-tama, pasien menuju loket administrasi. Di sana, pasien memberitahukan identitas dirinya kepada petugas administrasi untuk keperluan pencarian rekam medis. Jika pasien belum memiliki rekam medis, maka pasien perlu mengisi formulir rekam medis pada loket administrasi. Setelah itu, pasien mendapatkan nomor antrian. Saat nomor antrian pasien dipanggil, asisten dokter menuju loket administrasi untuk mengambil rekam medis, lalu menuju ruang pemeriksaan bersama pasien dan menyerahkan rekam medis tersebut ke dokter. Setelah proses pemeriksaan, pasien diberikan resep obat. Pasien kemudian menuju ruang tenaga kefarmasian untuk mendapatkan obat sesuai dengan resep yang diberikan. Asisten dokter menerima rekam medis pasien dan mengembalikannya ke petugas administrasi.

### **2. Apa kendala yang dialami klinik dalam proses pengelolaan rekam medis pasien?**

Jumlah pasien yang berobat tiap tahunnya semakin meningkat. Hal ini menyebabkan dibutuhkan tempat penyimpanan rekam medis yang lebih besar. Selain itu, pernah terjadi kejadian dimana rekam medis pasien tidak ditemukan, padahal pasien tersebut sebelumnya pernah berobat di klinik ini. Hal ini menyebabkan pembuatan rekam medis baru. Beberapa waktu kemudian diketahui bahwa rekam medis yang hilang tersebut ternyata diletakkan pada bagian awalan nama yang salah.

### **3. Informasi apa saja yang terdapat pada rekam medis klinik bapak?**

Di klinik ini, rekam medis pasien memuat informasi mengenai data diri pasien, seperti nama, nomor BPJS jika ada, tempat & tanggal lahir, dan nomor telepon. Selain data diri, terdapat informasi medis pasien, seperti tanggal pemeriksaan, dokter yang melakukan pemeriksaan, keluhan pasien, hasil pemeriksaan fisik, diagnosa, obat yang dibutuhkan, dan catatan tambahan.

### **4. Untuk pengelolaan obat, informasi apa saja yang diberikan?**

Cukup dengan nama obat, deskripsi dari obat tersebut, kemudian harga obat, dan jumlah obat yang tersedia pada klinik.

5. Apakah bapak memiliki kebutuhan-kebutuhan khusus yang harus tersedia oleh sistem?

Ada fitur untuk menyimpan, mencari, menampilkan, dan mengubah data pasien. Selain itu, ada fitur untuk menyimpan rekam medis ke dalam sistem, kemudian rekam medisnya juga bisa dicari, dicetak atau disimpan ke dalam laptop. Ada juga fitur untuk menyimpan, mencari, menampilkan, dan mengubah data obat. Tolong tambahkan juga fitur antrian pasien dan pengelolaan data pengguna.

6. Untuk fitur antrian, informasi apa saja yang diperlukan?

Nomor antrian pasien, nama pasien, nomor BPJS kalau pasiennya punya, tanggal mendaftar, jam mendaftar, dan jam pemeriksaan.

7. Siapa saja yang nantinya bisa menggunakan sistem ini?

Dokter, petugas administrasi, dan apoteker. Tolong tambahkan juga admin untuk bisa mengelola data pengguna sistem.